.EXE

APRIL 1990 VOL 4 ISSUE 10

Special Feature
NETWORKS

The Software Developers' Magazine

Programming C in comfort. We take a look at JPI's TopSpeed C.

User interface design. How to make your software easier to use.

How to write a .EXE file loader for MS-DOS.

How one developer conceived a new software package and brought it to market.

How to write assembly language add-in utilities for WordPerfect.

UNIX DEVELOPMENT & SUPPORT PERSONNEL

Introducing a major new UNIX enhancement...

QA Training now provides a complete range of UNIX courses with all the qualities that have made us Europe's leading independent technical training organisation.

Qualities which include our use of the very best materials, the highest calibre instructors, and considerable emphasis on practical hands-on sessions. For example, we provide *every* delegate on our support courses with their own central UNIX system.

QA UNIX courses can be 'standalone', or taken as a linked series. But whatever you choose, the objective is the same – to help you master the skills you need for optimum productivity in your work.

It's a unique, refreshing new approach to UNIX or AIX training that all developers and support personnel will find invaluable in their career development.

Select the course that suits you best from the panel opposite.

Now is the time to find out more and experience the difference that has made QA No. 1...

Return the coupon today or call Samantha Trinder on 0285 655888.

UNIX TRAINING COURSES

UNIX FUNDAMENTALS	Basic commands • The file system • The process environment • The shell • UNIX communications • Configuring the system • Basic systems administration
UNIX SYSTEMS PROGRAMMING	Code generation tools • Development tools • The source tree • UNIX and C • Process management • Pipes • Inter-process communication • Memory management
UNIX ADVANCED SHELL PROGRAMMING	Variables • Conditional flow control • Debugging shell procedures • Evaluation and quoting • Error handling • I/O redirection • Pattern recognition and substitution
UNIX SYSTEMS ADMINISTRATION	File system management • Back-ups • Process management • Security • Environments and profiles • Performance and tuning • Handling catastrophes
UNIX COMMUNICATIONS	Introduction to serial communications • The log in cycle • Introduction to the BNU • File transfer (UUCP) • Remote execution • Mail • Usenet • Products and Protocols

QA for UNIX

Please complete and mail to QA Training, Cecily Hill Castle, Cirencester, Glos GL7 2EF, England

☐ **Send me more details on QA's UNIX curriculum**

Other areas of interest: ☐ **OS/2** ☐ **DOS** ☐ **Windows** ☐ **Networks** ☐ **System Support** ☐ **C and C++**

NAME | TITLE

COMPANY

ADDRESS

TEL. NO.

QA TRAINING

CIRCLE NO. 830

Vol 4 Issue 10 .EXE April 1990

Editor
Robert Schifreen

Deputy Editor
Will Watts

Editorial Assistant
Wendie Leek

Senior Sales Executive
Sandra Inniss-Palmer

Group Recruitment Sales Manager
Jane Lyons

Recruitment Sales Executives
Jeremy Cox
Ed Butcher

Reader Services & Promotions
Helena Adams

Design & Layout
Katerina Adams
Raichel Affleck

Office Administrator
Carla Pearce

Cover Photography
David Pritchard

Publisher
Jon Howell

Editorial & Advertising: 081-994 6477
Subscription Enquiries: 044 282 4501

Editorial Policy
We aim to provide news, product reviews and technical features for those who develop PC software for both commercial sale and internal company use. Our policy is not to review any software product until it is available in its final form, in order to provide accurate figures on code size and speed. The Magazine welcomes articles from readers - please ask for our contributors' guide.

Subscriptions
.EXE is a magazine for managers and programmers involved in the development of software for the IBM PC and compatibles. .EXE is available only on subscription, which costs £35 per year (11 issues) in the UK. The Magazine is published on the 26th of the month preceeding the cover date. There is no January issue. International subscription rates can be found on the subscription card in this issue. Back issues are available at £3.50 each. To subscribe, call 044282 4501. We can invoice your company or take VISA or ACCESS cards.

.EXE Magazine is independent and not affiliated to any vendor of hardware, software, or services. Published by Process Communications Ltd, 10 Barley Mow Passage, London W4 4PH, England. Telephone: 081 994 6477 Fax: 081 994 1533 ISSN:0268-6872

Reacting to Readers
Our annual Reader Survey gives us an insight into the tools, methods, language and hardware that is being used, and that will be used in the process of software development.

We aim to keep in touch with software developers as much as possible. For this reason, we hold twice-yearly meetings at the .EXE offices and around the country, where readers can talk directly to the editorial staff and suggest ways in which the Magazine could be improved.

Support and Survive

During the course of the average week, I probably speak to at least a couple of dozen programmers. Some call the office, offering (or seeking) technical information. Some buy me dinner or a diet Coke. Some send me messages on Cix, and on other electronic systems that I use. Sooner or later, you can bet that the conversations get round to technical support.

Ask anyone in software development for their favourite support-related horror story, and the chances are that Microsoft will figure somewhere in the tale. Real Programmers lean over the table at dinner parties and say, 'I called Microsoft about an upgrade to C, and they kept me hanging on for a day and a half, and then never got back to me'.

Microsoft, one must concede, is a victim of its own success. It has such a large share of the PC language market (not to mention a 95% share in the operating system stakes), that there will always be a greater number of unhappy Microsoft customers than, say, Digital Research customers.

Providing support costs money, though few would deny that it's money well spent. Unfortunately, whatever you are willing to pay, it's very hard to find qualified support staff. It's widely regarded as being one of the least glamorous jobs in the industry. It does have its lighter moments, though. Everyone's heard the anecdote about the support person who, unable to diagnose a caller's problem, requests a copy of the floppy disk in question. Next morning, in the post, a photocopy of the disk arrives. This actually happened to a friend of mine, as did the occasion when he sent a novice user a disk, and reminded him to take the disk out the envelope before putting it in the drive. Novice user took a pair of scissors, and carefully removed the brown, circular disk from the black plastic envelope. After deciding that this disk was just too floppy to use, he called again.

It's a fact of life that dealing with this level of user is no fun for a technically qualified programmer. I spent 19 months doing it, and have no intention of doing it again.

Another problem, of course, is the low regard in which US software companies frequently hold their UK subsidiaries. Digital Research is unusual, in that a number of its products (DR DOS is just one) are developed solely in the UK. If a user has a problem, or if a developer (and we can be a pretty demanding bunch) has a very complex question, it's relatively easy for someone on DR's help desk to call the Hungerford office and ask someone to look up the solution in the source code. Microsoft (UK) Ltd, on the other hand, is little more than a marketing department. It holds hardly any source code of products. Combine this with the fact that most of the callers to its support desk are asking complex questions about complex products, and you begin to see the problem.

Of course, the solution is easy. First, UK software companies should hold all the source code, and everyone on the support desk should have a copy. Second, to keep customers happy, the support desk should answer calls from anyone who calls, regardless of whether they bought their copy of the product in the UK, or imported it from the US at half the price. If the US parent company wants to sell cheap software to UK residents, but won't give the UK arm the budget to support that volume of product, then so be it. Third, support staff should be paid at least £50K per year, and the number of staff on every support desk should be doubled. At least.

And, fourth, a worldwide ban on intelligent switchboards that play music on hold, and that 'can understand tone-generated information'.

.EXE

CLIBRARIES

DATABASE

Btrieve	OS/2	£400
Btrieve v5	MS-DOS	£170
Btrieve/Network	MS-DOS	£400
XQL	OS/2,PC-DOS	£540
Xtrieve Plus	OS/2,PC-DOS	£400
C/Database T'lchest sce	PC-DOS	£ 39
CBTREE	(source any C)	£135
C-Index/Plus	(source any C)	£175
C-ISAM (L,MS)	MS-DOS	£220
C-tree	(source any C)	£230
Essential Btree (s'ce)	PC-DOS	£120
R-tree	MS-DOS	£170
db-VISTA III s'gle user	PC-DOS	£575
db-Query v2 s'gle user	PC-DOS	£575
Lattice dBCIII (L,MS)	MS-DOS	£165
Lattice dBCIII+ (L,MS)	MS-DOS	£385
SftFcus Btree&Isam (sce	any C)	£ 80
Virtual Memory File Man.	PC-DOS	£ 80
Zortech Btree/ISAM	PC-DOS	£ 75

GRAPHICS

Enhanced Graphics Tlkt	PC-DOS	£170
Essential Graphics v3	PC-DOS	£235
GFX Graphics	PC-DOS	£ 75
GraphiC v5 (CI,L,DS,MS)	PC-DOS	£290
Graphics Menu (MSC,TC)	PC-DOS	£105
GSS Kernel system	OS2,PC-DOS	£350
GSS Graph Dev T'kit	OS2,PC-DOS	£350
GSS Graphic Plotting sys	PC-DOS	£245
HALO '88 (MSC5,LAT,TC)	PC-DOS	£230
HGraph (MSC)	PC-DOS	£ 80
Ingraf (MSC)	PC-DOS	£175
MetaWINDOW (CI,L,MS,T)	PC-DOS	£145
MetaWINDOW/Plus	PC-DOS	£190
MetaWINDOW/Premium	PC-DOS	£315
TurboWINDOW/C (Turbo)	PC-DOS	£ 75
QuickWINDOW/C (QC)	PC-DOS	£ 75

SCREEN & WINDOWS

Vitamin C (MS5.1)	OS/2&PC-DOS	£240
Power Screen (MS&TC)	PC-DOS	£105
Blaise View Mngr.(s'ce)	PC-DOS	£245
Lattice Curses	PC-DOS	£105
Entelekon Windows(s'ce)	PC-DOS	£105
Multi-windows (MS,L)	PC-DOS	£220
Panel Plus II (source)	PC-DOS	£270
Vitamin C (source)	PC-DOS	£165
Vermont Views (MSC,TC)	PC-DOS	£345
Greenleaf Datawindow	PC-DOS	£180
Greenleaf Makeform	PC-DOS	£ 60
C-Scape(+s'ce&Look&Feel)	PC-DOS	£330
Zortech Windows(ZC++ZC3)	PC-DOS	£ 39
Curses/PC w. s'ce	PC-DOS	£265

GENERAL FUNCTIONS

C+0 Class Lib (MS,TC)	PC-DOS	£155
C+0 Cl's Lb (MS)WINDOWS&	PC-DOS	£155
C/Utilities T'chest sce	PC-DOS	£ 39
CQL s'ce	PC-DOS	£270
C-Worthy Int.Lib (L,MS)	MS-DOS	£175
Code Runner	PC-DOS	£105
D-Tree (source)	PC-DOS	£345
Greenleaf Functions s'ce	PC-DOS	£105
G'leaf Super Functions	PC-DOS	£135
G'leaf Business Mathlib	PC-DOS	£180
Smorgasbord C-food	PC-DOS	£ 65
C Tools Plus/6.0 (MS)	PC-DOS	£105
Power Search	PC-DOS	£105
Turbo C Tools v2	PC-DOS	£105
Essential C Utils s'ce	PC-DOS	£145
Entelekon Funct(source)	PC-DOS	£105
Pro-C & Workbench	PC-DOS	£350
Resident-C (MS,L)(s'ce)	PC-DOS	£145
The Heap Expander(MS&TC)	PC-DOS	£ 65
Zortech Supertext	PC-DOS	£ 39

COMMS LIBRARIES

C Asynch Manager V3s'ce	PC-DOS	£135
Essential Comms(L,MS,T)	PC-DOS	£195
Greenleaf Comms(source)	PC-DOS	£135
Multi Comms (L,MS)	PC-DOS	£165
Silvercomm C Asynch	PC-DOS	£145
Zortech Comms (QC&TC)	PC-DOS	£ 65
Zortech Comms(ZC++&ZC3)	PC-DOS	£ 65

SCIENTIFIC LIBRARIES

Wiley Scientif.Lib.v2	ANY C	£155
Mathpak 87 (L,MS)	MS-DOS	£ 85

PROGRAMMERS UTILITIES

PC-Lint	OS/2,MS-DOS	£ 85
C-Documentor	PC-DOS	£195
C-Scan	PC-DOS	£195
Clear+ for C	PC-DOS	£165
Lattice Comp Companion	MS-DOS	£ 65

REAL-TIME & MULTI-TASKING TOOLS

Concurrent C (PC/MPX)sce	MS-DOS	£ 60
Multi C (L,MS,CI,TC)	PC-DOS	£165
Op.Sys.T'box (MS) s'ce	PC-DOS	£ 65
Timeslicer v3.01 (L)	PC-DOS	£175
Timeslicer v5 (MS)	PC-DOS	£175
Over-C (L,MS)	PC-DOS	£225

ASSEMBLERS

MS Macro-86 v5.1	OS/2&PC-DOS	£ 95
PharLap 386 ASM.LINK	MS-DOS	£400
PharLap 386 ASM/Linkloc	MS-DOS	£565
PharLap 386/VMM	MS-DOS	£240
Optasm	MS-DOS	£ 95
Turbo Assembler/Debugger	PC-DOS	£105
2500AD Z80 ASM	CP/M-80	£ 70
Dig.Res. RMAC	CP/M-80	£180
SLR Z80ASM	CP/M-80	£ 40
SLR Z80ASM-PLUS	CP/M-80	£140
SLR MAC	CP/M-80	£ 40
SLR MAC-PLUS	CP/M-80	£140
SLR 180 (Hitachi)	CP/M-80	£ 40
SLR 180-PLUS (Hitachi)	CP/M-80	£140

Not all assemblers are supplied with a linker. Check before ordering.

PROLOG LANGUAGE

Arity Standard	PC-DOS	£ 79
Arity Prolog Int. v5.1	PC-DOS	£225
Arity Int. & Comp. v5.1	PC-DOS	£495
Flex Expert Dev Sys	PC-DOS	£1210
LPA PROLOG Prof v3	PC-DOS	£755
Micro-PROLOG v3.1	MS-DOS	£ 70
Prolog-86 plus v1.0	MS-DOS	£195
Prolog-2 Personal	PC-DOS	£145
Prolog-2 Programmer	PC-DOS	£495
Prolog-2 Professional	PC-DOS	£995
Turbo-Prolog v2	PC-DOS	£ 79
ADA Educ.Prolog	MS-DOS	£ 26
ADA FS Prolog	MS-DOS	£ 45
ADA VMI Prolog	MS-DOS	£ 70
ADA VML Prolog	MS-DOS	£150
Micro-Prolog v3.1	CP/M-86	£ 70
Micro-Prolog v3.1	CP/M-80	£ 70
Prolog-1 v2.2	CP/M-80	£225
LPA Mac Prolog v3 Apple Mac Pls		£450

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, ISIS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, DEC RT- 11, IBM BEF, ATARI ST, AMSTRAD, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

PRICES & DELIVERY

Prices do not include VAT or other local taxes but do include delivery in UK and Europe. Please check prices at time of order, ads are prepared some weeks before publication.

This page lists some of our products. Call us for a complete pricelist.

Order by phone with your credit card.

PROGRAMMING TOOLS

Ada Compilers	Algol Compilers
Assemblers & Libs	AWK
Basic Compilers	Basic Interpreters
Basic Utilities	Basic Libraries
BCPL Compilers	C Compilers
C Interpreters	C Libraries
C Utilities	Cobol Compilers
Comms.Libraries	Cross Assemblers
Database Libs.	Debuggers
Dis-assemblers	Editors
Engineers Libs.	Expert Systems
Forth	Fortran Compilers
Fortran Libraries	Graphics Libraries
Icon	Linkers/Locaters
Lisp	Modula-2
Nial Interpreters	OPS 5
Pascal Compilers	Pascal Libraries
Prolog	Rexx
Risc	Screen Libraries
Smalltalk	Snobol

We stock many items for which there is no space in these advertisements.

THE C LANGUAGE

High C V1.6 has been considerably improved, with better Microsoft C compatibility, and new documentation

C COMPILERS

Avocet C	PC-DOS	£210
Aztec C86 Professional	MS-DOS	£120
Aztec C86 Developer	MS-DOS	£175
Aztec C86 Commercial	MS-DOS	£290
CI C86 Plus v1.3	MS-DOS	£290
ECO-C88 V4.0	MS-DOS	£ 70
HIGH C (Metaware) v1.6	MS-DOS	£420
HIGH C 386 v1.6 PL386&	MS-DOS	£635
Hi-Tech C	MS-DOS	£135
Lattice C v6	OS/2&MS-DOS	£175
Mark Williams LETS C v4	MS-DOS	£ 60
Microsoft C v5.1	OS/2&PC-DOS	£250
Microsoft QuickC v2	PC-DOS	£ 60
M'soft QuickC/Quickasm	PC-DOS	£105
MIX C	MS-DOS	£ 20
MIX Power C & lib s'ce	PC-DOS	£ 29
Turbo C v2	PC-DOS	£ 79
Turbo C Professional v2	MS-DOS	£160
Topspeed C for OS/2	OS/2	£335
Topspeed C (std)	PC-DOS	£135
Topspeed C Professional	PC-DOS	£270
Watcom Express C v7	PC-DOS	£ 65
Watcom v7 opt comp + tls	PC-DOS	£250
Watcom C/386 PL386&	MS-DOS	£625
Zortech C v3	PC-DOS	£ 45
Zortech C++ v2	PC-DOS	£120
Z'tech C++ v2 Developer	PC-DOS	£270
Aztec C86 Developer	CP/M-86	£245
Aztec C86 Personal	CP/M-86	£140
Hi-Tech C	CP/M-86	£135
Lattice C	CDOS	£395
HIGH C (Metaware)	Flex OS 286	£420
Avocet C	CP/M-80	£210
Aztec C Personal 1.06D	CP/M-80	£120
Aztec Commercial 1.06D	CP/M-80	£200
Eco-C w. Lib s'ce Z80 +	CP/M-80	£ 60
Hi-Tech C Z80 +	CP/M-80	£ 99
Mix C Z80 +	CP/M-80	£ 35
Toolworks C/80 v3.1	CP/M-80	£ 45
Aztec C65 V1.05	Apple DOS	£200
Aztec C65 V3.20	Apple PRO-DOS	£245
Aztec C68K MPW C	MACINTOSH	£110
Aztec C68K C	MACINTOSH	£ 75
Aztec C68K/AM Developer	AMIGA	£175
Aztec C68K/AM Prof'snl	AMIGA	£120
Aztec C68K/ST Developer	ATARI	£175
Aztec C68K/ST Prof'snl	ATARI	£120
Hi-Tech C	ATARI	£ 99
Lattice C	ATARI	£ 85
Mark Williams C v2	ATARI	£110
Laser C	ATARI	£135
Prospero C	ATARI	£105

C CROSS COMPILERS

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020. Please call for information or advice.

C INTERPRETERS

Instant C/16M supports large programs on 286/386 machines.

C-terp V3.0X	PC-DOS	£165
Interactive C	PC-DOS	£195
Introducing C	PC-DOS	£ 85
Living C Plus	PC-DOS	£135
Instant-C v3	PC-DOS	£380
Instant-C/16M	PC-DOS	£680
RUN/C Professional	MS-DOS	£110

ADA COMPILERS

The new Ada Student is an affordable way to learn this important language.

Ada Student	PC-DOS	£ 75
Janus/Ada compiler	PC-DOS	£ 355
IntegrAda	PC-DOS	£ 645

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

GREY MATTER
2 Prigg Meadow, Ashburton, Devon TQ13 7DF
TEL. (0364) 53499

News

Extended Family

Lattice is to release two new versions of its 80x86 C compiler. The Lattice 80286 Development System includes a royalty-free DOS Extender, which developers may give away with their software. The Extender gives MS-DOS programs access to 16 MB of address space. The compiler runs under DOS, Extended DOS or OS/2 and can produce code for any of these environments. Lattice has also invented what it calls 'Extended Family Mode', under which the same compiled program can run under all three OS environments without modification. The software should be available by the time you read this.

Lattice's 80386 C Development System runs in an even more exotic range of environments. Like Metaware's and Watcom's similar products, it supports Phar Lap's 386 Extender, Lattice's own 386 DOS Extender and - here's the big one - OS/2 V2.0. This is the version of OS/2 that has yet to see general release. What with Microsoft's own 32-bit C compiler still not generally available, Lattice may well have brought off something of a coup. We are slightly alarmed by the official release date of the product, which is the 1st of April.

US prices for the products are $495 for the 286 compiler and $900 for the 386 version. The UK distributors are Roundhill Computer Systems; you can contact them on 0672 54675.

Eiffel comes to Britain

The Eiffel object-oriented language differs from most of the other OOP players in that it is a comparitively 'pure' implementation of OOP; for example, it is not a superset of an eccentric unorthogonal language like C. On the other hand, it is supposed to be much more practical than the esoteric, but cumbersome, Smalltalk language. Like AT&T's implementation of C++, it generates C language as its object code, so is extremely portable. It was invented by Dr Betrand Meyer, a very influential figure in the OOP world, who markets it through his US company Interactive Software Engineering.

Eiffel has now acquired a UK distributor, Applied Logic Distribution, who will also provide technical support and training. The language is available on a variety of platforms, from 386 XENIX, through Sun workstations up to mainframes. The entry level price is around £1000. For more information, please contact Grant Price at ALD (01 780 1088).

Bounds Checker

Nu-Mega Technologies, who make the Soft-ICE debugger, are about to launch two new products. Bounds-Checker uses the hardware debugging features of the 386 and 486 chip to monitor your DOS system. If your program attempts to write to an area of memory which it's not allowed to (like the operating system, for example), up pops the offending piece of source code, so you can see what's going on. Bounds-Checker finds its way around the source code by interpreting the CodeView debugging information files produced by Microsoft compilers.

Also about to be released is NLM-Check, which is a Network version of Bounds-Checker, with the added capability of analysing the execution of your code in real time, and telling you which routines are hogging CPU time. Again, you are shown the offending source code. The products cost $249 for Bounds-Checker and $499 for NLM-Check . Phone Nu-Mega, in the US, on 0101 603 888 2386.

CASE:PM Upgraded

In last November's issue of .EXE we reviewed a handful of tools that aid the development of Presentation Manager applications. One of these, CASE:PM, is now up to version 2. The new release supports the generation of multiple windows, and you can now write your own routines in C and link them into the application that CASE:PM generates. There's also a new message browser facility - point at the PM messages that you want your application to respond to, and fill in the code for the responder.

The in-built prototyper has been upgraded too. A new edit facility allows cutting, pasting and undoing. When you want to see how your completed application will look, you can run the prototype on screen without having to compile a line of code.

By the time you read this, both versions of CASE:PM will be available. The one with a C code generator costs £1295, and the COBOL version is £1595. Call the UK distributor, QA Training, on 0285 65588 for details.

CASE:PM 2.0 needs OS/2 1.2, and an OS/2 SDK from Microsoft or IBM.

Meet Bjarne

Bjarne Stroustrup, progenitor of C++ and part-time .EXE interviewee, is to be the keynote speaker at a one day C++ Technical Seminar to be held in London on 12th June. Dr Stroustrup will discuss details of Release 2 features of the language and describe some intended future developments. More details from Nicky Wheeler at AT&T USO Europe, on 01 567 7711.

Grabber

The Video Frame Grabber, from Rombo (0506 414631) is a £150 card that plugs into your PC, and lets you play your video recorder through the computer. When you see a frame you like, hit the button, and a GEM, TIFF or PCX format file will appear on your hard disk.

Flush subbies

A survey of the takings of computer subcontractors suggests that an IBM programmer, with three years' experience, can expect to earn, on average, £28,500 in his first freelancing year in the UK regions. If he comes to the Metropolis, this goes up to £31,600; but if he goes abroad, it's jackpot time at nearly £40,000.

Hot and FAST

FAST, the Federation Against Software Theft, has set up a free hotline, where you can report allegations of software misuse and piracy. A similar line in the US which has been running for some months averages between 10 and 20 calls per day. If you suspect a UK company of illegally copying software, call 0800 181 502. They'll be waiting.

Fast Firefox

The Trailblazer Communications PowerPack is an MS-DOS and OS/2 file software comms suite bundled with a high-speed modem. The manufacturer, Firefox Communications of Berkshire, claims that the software can achieve 40,000 - 50,000 bps in file transfers, using a special protocol, named Hyper-Protocol, which can switch its compression techniques on-the-fly. The package retails at £1495, and is available from your dealer.

Code Base 4

Code Base 4 is a C library which can access dBASE III/IV compatible data, index and memo files. It is written in ANSI C, so is compatible with MS-DOS, OS/2 and UNIX, with over 200 function calls. The distributor, The Software Construction Company (0763 73455), claims that it accesses data five times faster than dBASE IV. The software retails at £185.

Prolog++

The UK company Logic Programming Associates has produced a new OOP programming language for MS-DOS computers: Prolog++. The new product multiple-inherits (geddit?) characteristics from both Prolog and declarative object oriented languages such as CLOS, the LISP derivative. There is support for all of the usual OOP mechanisms, including, as it happens, the controversial multiple inheritance facility. To give you a feel for the language, here is a little piece of code used to define a clock object, taken from a simulation of the behavior of customer queues within a bank.

```
%_Generic clock object has a current time, a stop time
% and the ability to increment current time by one tick.
open_object clock.
now is 1.
stop is 100.
tick :-
 writeseqnl([ 'End', of, tick, myself::now ]),
 nl,
 ( subs(teller), subs(customer_queue) ) <- after_tick,
 myself::now += 1.
close_object clock.
```

LPA is selling Prolog++ in two versions: a Programmer's Edition (£495), which includes an incremental compiler and source level debugger, and the Developer's Edition (£1500), which adds an optimising compiler, a mixed language interface and a library of WIMP objects. This PC version of Prolog++ follows a Mac version released last year. Call LPA on 01 871 2016 for details.

Norton Backup

Peter Norton Computing is now shipping the Norton Backup, an MS-DOS utility for backing up and restoring hard and floppy disks. Claimed special features include network compatibility, four levels of data compression, and something called Fastpath technology. This, it says here, 'includes dual-DMA disk writing techniques, advanced skewing control and optimised data compression. Fastpath automatically configures itself to the user's hardware'.

For a product as critical as a backup utility, I have to admit that I'd be rather wary of using the first release of this program. A couple of years ago, when Norton first released a file de-fragmenter, it managed to scramble a large part of my hard disk. A colleague's machine suffered the same fate. Perhaps, in the interval, the Norton beta test team has been upgraded. Norton Backup retails at £149.

Matrix Layout 2

After months of delay (we were originally promised a copy in the middle of last year), we have finally seen a copy of Matrix Layout V2.0. This is a code generator for the PC. You click on menu options to build up a flow chart of what you want your program to do. Function calls are represented by 'black boxes', and a small library of functions is included. Those that aren't included, you can write yourself in C or assembler, and add them to the Matrix desktop with the toolkit supplied. Once your application is finished, click a menu, and the system generates C, BASIC or Turbo Pascal source code.

Version 2 comes with a number of enhancements, including 'black boxes' for dBASE, which allow Matrix programs to read and write dBASE files. Details from Matrix Software, on 0752 796363, or from Stand 5680 at the Which Computer? Show, which takes place at Birmingham's NEC from 24th to 27th April.

Shareware Catalogue

From time to time, we mention Shareware software. Shareware is not 'free' software. The author of the program allows you to copy and distribute it freely, but if you like it, you are supposed to register it by sending some money to the authors. If you don't use the program, you just reformat the disk.

Shareware normally circulates on bulletin boards. However, with less than 10% of the country's PCs linked to modems, several companies have made money by distributing shareware, charging a fee (up to £10 per disk) for duplication and postage. Of course, they can't charge for the software itself, because the author forbids it.

PC Serve is a Shareware library, and it has just produced an updated catalogue of PC shareware. In among the games and the Lotus macros, there are a few pages of interest to .EXE readers, including languages (BASIC interpreters, C interpreters and compilers, Tiny Pascal, lots of cross assemblers, C libraries) and lots of dBASE files. For a copy, call 01 864 2611.

New 486 Server
Like many other manufacturers, HM Systems will be using the Which Computer? Show to launch its 486-based machine. HM Systems will be showing their network server, with 10 free slots, and 3 GBytes of hard disk. If you can't make the show, call 01 209 0911.

Z80 C
Dynamic C is a PC-based C cross-compiler which can generate code for the Zilog Z80 and the Hitachi 64180. It includes an integrated environment, with a text editor and debugger - the latter requires a communications link to the target system in order to work. The distributor is Nohau UK: 0962 733140.

QuickCopy
Disk copying is a time-consuming, dreary job, especially now that dual floppy drive machines are relatively uncommon. Kwikcopy, from Ctrl Alt Deli (0908 662759), claims to be an intelligent copier. It can spool data to hard disk, skip blank tracks and make multiple copies of the same master without rereading it. The software costs £49.00, or £59.00 for a souped-up version.

dBASE Seminar
The dBASE Users Group is holding a seminar in London on 19th April. The theme of the occasion is to look at ways of linking dBASE databases with other hardware and software, such as statistical packages and bar code readers. Call Anne Boreham, the Group's Administrator, on 0256 768646 to hear more.

Recognised Pattern
If you are an academic - anything from graduate student to Head of Department, and not necessarily specialising in Computer Science - then you can apply for a fellowship at the Research Initiative in Pattern Recognition, with the chance of spending 6-12 weeks at the UK's leading research establishment in this field, RSRE Malvern. To make an application, send a one page outline of your proposed research topic to Paul Gregory, Director, RIPR, St Andrews Rd, Malvern WR14 3PS.

NetOp Upgraded
NetOp is a utility that lets you log on, remotely, to any PC or server on a network. By the use of some clever windowing routines, you can effectively use all the PCs on a 10-station network at the same time, and all from the same machine. If this sounds like the answer to your troubleshooting needs, call Richmond Systems Ltd on 01 940 6986.

Needs Clipping

```
plusyear = 0

IF (year_diff > 0 .AND.
   month_diff = 0 .AND. day_diff >= 0);
 .OR. (year_diff > 0 .AND. month_diff > 0)
 plusyear = 1
ELSE
 plusyear = 0
ENDIF

IF plusyear = 1
 age = (YEAR(DATE()) - YEAR(born_date))
 ELSE
 age = (YEAR(DATE()) - YEAR(born_date)) - 1
ENDIF
```

We are grateful to Mr Ian Butterworth, of QBS Ltd, for drawing our attention to this piece of Clipper code, published on page 20 of the Sept/Oct '89 edition of Nantucket News and reproduced here without permission. Hidden somewhere in this code fragment, there is an opportunity for optimisation. Can you spot what it is? A free subscription to anybody who sends us a demonstrably more flatulent piece of code, extracted from any recent computing journal other than this one (we're not interested in the plank in our own eye), coded in any language which our editorial staff can grasp. Entries should be sent for attention of The Editor, at the address given on Page 2.

Better overlays

Turbo Pascal V5.5 has its own overlay system, but it has some important limitations. It is not possible to swap out data - this must all be kept in primary memory. It is difficult to profile an overlaid application, and therefore hard to optimise it. Turbo VMM, from Danish software house Syntax, provides an alternative to Borland's bundled overlay unit which addresses these problems. It lets you swap out both code and data to expanded memory, extended memory or disk. The call interface is very similar to the 'real' overlay unit; simply by changing the `OvrInit` call to its `TVMMInit` equivalent, existing code can be adapted to use Turbo VMM, although to access the more advanced features, further modification is required.

Turbo VMM is priced at $129 US for the binary version, or $179 with the (Turbo Assembler) source code. Contact Syntax on 010 45 3131 9299.

File Transfer

Trax is a remote file transfer facility. You run it on the machine that contains all your important files. Then, with the aid of a modem and *another* copy of Trax (spot the clever marketing strategy to increase sales), you dial up the remote machine and copy files from one to the other. If you're worried about security, you can password-protect the machine, and allocate separate IDs and privilege levels to all callers. If you're not sure of the name of the file you want, you can even browse it before starting the transfer.

Trax costs £125 per copy, or £195 for a pair. Don't even think about buying one copy and using it at both ends of the link - the system will refuse to transfer files between two copies of Trax which bear the same serial number. The UK distributor is Internation Data Security, who can be reached on 01 631 0548.

Now we have ANSI

Compiler vendors who describe their product as'100% ANSI compatible' had better look sharp, according to Neil Martin of the BSI. Now that the ANSI standard officially exists, any vendor who claims ANSI compatibility, but whose product falls short in some way, might find himself hauled up in front of the Trading Standards Authority, or even in court. In the future, he may even fall foul of EC regulations. The vendor's main line of defence would be to produce a formal BSI validation certificate - the BSI is to start testing and awarding these in June.

The BSI is also to start marketing Knowledge Software's Model Implementation of C. This is a C pcode-style interpreter which not only conforms strictly to the ANSI Standard, but also detects any dependence in the code on implementation-defined or undefined behaviour. The idea is this. You create your intended portable code on an ordinary ANSI C compiler. When it appears to work, you run it on the model implementation, which highlights any areas where you have accidentally relied upon the idiosyncrasies of your compiler - for example, expecting the type `char` to default to unsigned (ANSI says that it can default to unsigned *or* signed). The compiler also picks up things like array bound violations and the use of uninitialised pointers. At £10,000 for the C source code it's not a cheap toy - it is intended for corporates who need to create bulk quantities of code. Perhaps a certain spreadsheet manufacturer, whose PC product seems to be appearing on all sorts of machines these days, might be interested.

When I tried to give out the BSI's number last month, I hit a fax; so here, to make amends, are two numbers: 0908 220908 and 0908 220022.

WordPerfect Bug

We came close to losing a page of news this month. The Editor uses WordPerfect, the latest version (5.1) which has a bug: if you call up the spell checker, as the program is about to perform an automatic timed backup, the system erases your backup files and hangs. It performed this trick at a critical moment, this month. Next month's news will come to you courtesy of EDLIN.

ANSI C Compiler

Microtec Research has released an ANSI-compatible C compiler for the 68000 family. The product includes additions that, says Microtec, make the product ideally suited for use in real time or embedded systems . The code generator produces 68000 code, the compiler runs on a PC, Sun, DECstation or VAX. Details from Microtec, on 0256 57551.

Pascal Browse

Symantec has released version 3.0 of its Macintosh compiler Think Pascal. The new product has its own class library, support for Apple's MacApp and a class browser - vital for navigating through large programs. Think Pascal requires at least one megabyte of RAM, much more if the class libraries are to be used. It costs £150, and is available from dealers.

Take your Pick

If you use the Pick operating system, then you may be interested in the third edition of the Pick Resources Guide/International, which has just been published. It is a directory of Pick-based hardware, software, training courses and expertise. The price of the guide is £39.95 to the Pick community. The publisher is ALLM Systems and Marketing (0923 30150).

Pascal to C

Migration Technology, known for its BASIC-To-C translator, has spread its wings and produced a Pascal-To-C program. It is mostly seen as an aid to porting Pascal programs onto UNIX platforms, although MT suggests that you could also use it as a development tool, to save retraining Pascal programmers, which seems rather far-fetched. MT's number is 0628 668 511.

Fax Help

The PC Independent User Group has started offering a fax hotline service. Members can fax printer problems or screen dumps for expert perusal, and receive copies of relevant technical articles and notes in return. Annual membership of the group costs £29 for businesses: contact it on 0732 771512 (voice) and 0732 771513 (fax).

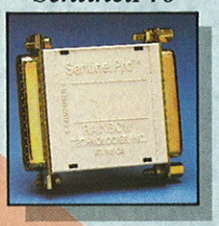
.EXE

S O F T W A R E S E C U R I T Y

With £1 Billion Worth Of Protected Software...

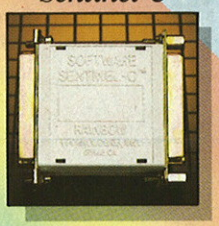
Rainbow Is The Safe Bet.

SentinelPro™



• Runs under DOS, OS/2 and Xenix • Algorithm technique (Never a fixed response) • External parallel port installation • Minimal implementation effort • Higher level language interfaces included • 100 times faster than fixed-response devices (1 ms) • ASIC design for reliability

Sentinel-C™

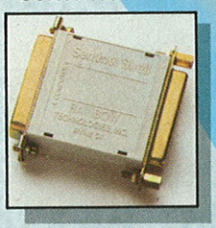


• Protects multiple packages with one device • 126 bytes of non-volatile memory programmed before shipment of the software • Rainbow supplies a unique adapter for programming the unit • Higher level language interfaces included • Runs under DOS, OS/2 and Xenix • External parallel port installation

Eve™

• For the Macintosh SE and II • Complies with Apple Desktop Bus Interface requirements • Rainbow-assigned developer passwords to prevent tampering by other developers or sophisticated "hackers" • 7 locks per key, usable individually or in combination, on one or up to seven applications

SentinelShell™



• Runs under DOS on IBM PCs and compatibles • Protects without requiring access to the source code • Completely transparent to the end user • User-friendly software • Pocket-size key attaches quickly to any standard PC parallel port • ASIC design for reliability

DataSentry™

• Completely user-installable • Pocket-sized external device • Menu-driven, user-friendly interface • Single- or multi-user security system • Audit trail, log-on identifiers and automatic encryption/decryption of entire directories • Secures data transmitted by modems • Prevents recovery of data by utility programs

Software developers creating the latest applications for the IBM PC/XT/AT, PS/2 and compatible systems can now turn to the Software Sentinel range of hardware keys for the *first-class* in *world-class* software protection.

There's the best-selling SentinelPro, known worldwide for its virtually unbreakable security, its ASIC technology and its invisible operation.

A close relation, the Sentinel-C for custom configurations, enables multiple package protection with a single device.

For the Apple market, security-minded Mac software developers can now secure their return-on-investment, too. Eve plugs into the Mac ADB connector and is completely transparent to the user–providing up to seven programmable security locks per key.

Rainbow's latest protection strategy is the SentinelShell–that lets users place a "shell" around existing, off-the-shelf programs. Access can be limited to those issued a key, so that libraries, educational establishments and corporations can very simply guard their investments.

Available soon from Rainbow is the DataSentry, a user-installed key that provides low cost security for sensitive data in both database applications and corporate/banking environments.

No matter where you sell your software worldwide, stay in control of your distribution and revenue by choosing the internationally accepted standard in protection... Rainbow Technologies. Be sure. Protect your pot of gold at the end of the rainbow.

CALL TODAY OR ORDER NOW.

☐ Please send me a SentinelPro Evaluation Kit.
I enclose £50 + VAT Payable to Rainbow Technologies Ltd.

☐ Please debit my credit card. Access ___ Visa ___ Amex ___
Expiry Date _____ (PLEASE TICK)

☐ Please send me more information.

Name _____

Position _____

Address _____

Telephone _____

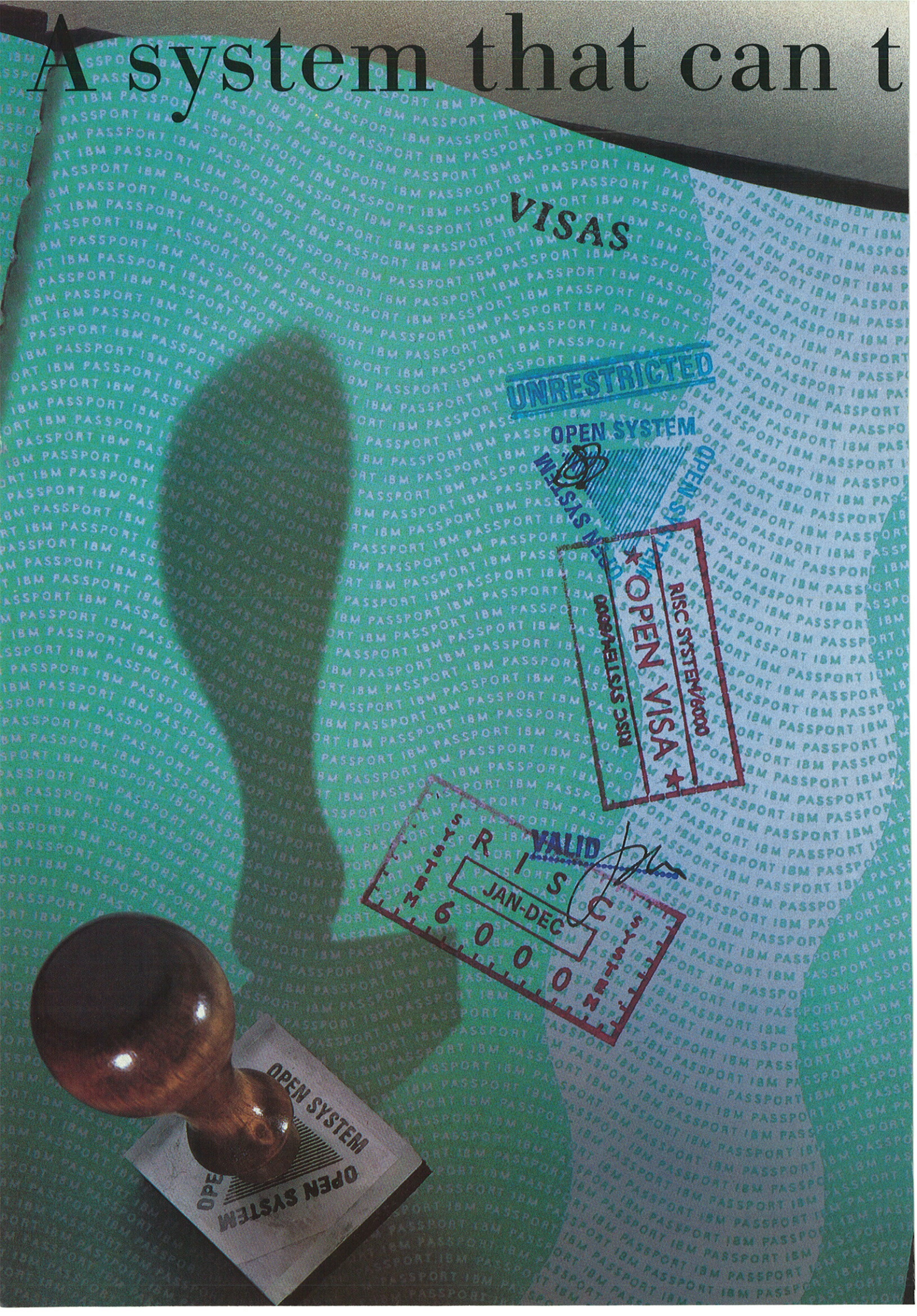
Signature _____ *.EXE 4/90*

RAINBOW TECHNOLOGIES LTD., Shirley Lodge, 470 London Road, Slough, Berkshire SL3 8QY.

RAINBOW TECHNOLOGIES

Shirley Lodge, 470 London Road
Slough, Berkshire SL3 8QY
Tel: 0753-41512 Fax: 0753-43610

CIRCLE NO. 832


A system that can t
VISAS
UNRESTRICTED
OPEN SYSTEM
OPEN VISA
RISC SYSTEM/6000
VALID
RISC
JAN-DEC
SYSTEM/6000
OPEN SYSTEM

ake you anywhere.

RISC System/6000, from IBM.

The power of the world's finest computers has never been more accessible.

It's an open system. AIX, IBM's implementation of the UNIX* operating system, gives you access to thousands of multi-user solutions. The potential is almost unlimited, both now and in the future.

A flexible system. For commercial multi-user applications, as a high performance workstation, or as a file-server on a network.

A growing system. A simple fact: RISC System/6000 accommodates growth. With the AIX base you can easily add users or change entire networks, so the system adapts to you: whatever your business requirements.

A cost-efficient system.

You'll pay less for RISC System/6000 than for anything comparable. And that's a fact.

Also, you'll be investing in an IBM system. That means outstanding reliability and technical leadership, with our specialist RISC System/6000 remarketers providing solutions and support.

When you're making decisions about your company's future, the message is clear.

Don't gamble.

Put your money on RISC System/6000.

It's open. It's faster. It's affordable. It's IBM.

For details of the revolutionary RISC System/6000, 'phone Peter Mercer on 01 995 7700 during office hours. Or write to him at the National Response Centre, IBM United Kingdom Limited, FREEPOST, London W4 5BR.

IBM

"I think, therefore IBM."

CIRCLE NO. 833

*UNIX is a trademark of AT&T in the USA and other countries.

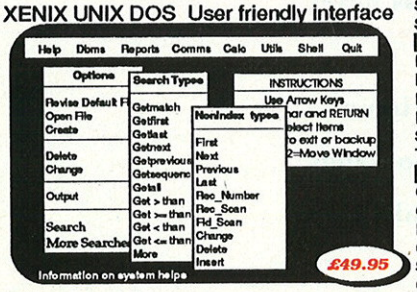
Leading Product Prices ~~Slashed~~

XENIX UNIX & DOS Users

Release the POWER of your UNIX XENIX Operating System

MENU - IX

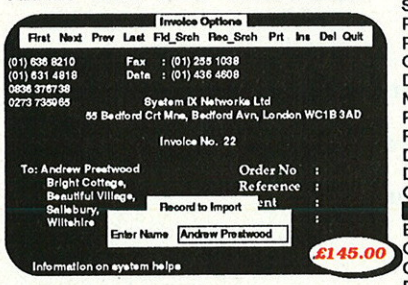
XENIX UNIX DOS User friendly interface



Powerful, friendly, colourful, complete menu system that *you* can customise to exacting requirements.

SYS - IX

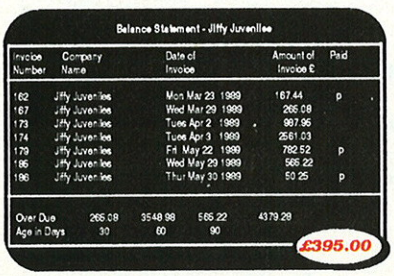
XENIX - UNIX - DOS - 4GL Database



A powerful & complete applications generator for forms & database management, inlcudes Menuix.

SYSTEM - IX

XENIX - UNIX - DOS - 4GL Reporting



Powerful toolset for reporting database file systems under Menuix & Sysix

Special Offer on Sys-IX

Free Application Modules

Bank Reconciliation
Integrated Customer Name & Address
Invoice & Ledger Database

Free Report Modules

Monthly turnover
Monthly VAT, monthly gross profit
Auto statement production with
Outstanding aged analysis

Free Utilities

General User Reporting Utility
Desktop Calculator & Calendar
Artificial Intelligent Help System
Programs is designed to enable
you to use your own stationary.

Prices exclude VAT & Delivery

SAGESOFT	
Sterling Accountant	£169
Sterling Accountant Plus	£279
Sterling Bookeeper	£75
Ster. Financial Controller	£349
Ster. Job Costing	£150
Ster. Network Accountant	£270
Ster. Net. Accountant Plus	£365
Ster. Net. Fin. Controller	£495
Ster. Net. Job Costing	£185
Ster. Network Payroll II	£185
Ster. Payroll II	£145
CAD & DESIGN	
Intro Training Course	£75
Drafix 3D Modeler	£247
Drafix Ultra	£475
Report Writer	£135
Symbol Lib Arch/Elec/Mech	£125
Turbo CAD v1.5	£75
COMMUNICATIONS	
Crosstalk For Windows v1.0	£95
Crosstalk XVI v3.62	£99
PC Anywhere III v3.10	£99
Carbon Copy Plus v5.1	£119
Smartmove 470	£157
Chit Chat Dual	£79
DATABASE SOFTWARE	
dBase III Plus v1.1	£325
dBase IV	£375
dbase IV Devel Edition	£599
Sys-IX Xenix/Dos/Unix	£195
System-IX Xenix/Dos/Unix	£395
R-Base Dos v2.11	£377
R-Base Tools	£79
Clipper Summer 87	£325
Documentor	£193
Nantucket Tools	£155
Retrieve III Single User	£119
Professional File v2.0	£219
DataPerfect Sngl N/W v2	£347
DBXL v1.3	£240
QuickSilver v1.3	£390
MICROSOFT	
Basic Compiler v6 OS/2	£169
C Compiler v5.1 OS/2	£199
COBOL Compiler v3.0	£388
Fortran Compiler v5 OS/ 2	£219
Macro Assem v5.1 OS2	£95
Pascal Compiler v4 OS/2	£146
Quick BASIC v4.5	£65
Quick C v2.0	£65
Quick C & Assemb v2.01	£102
Quick PASCAL	£65
Windows 286 v2.1	£65
Windows 386 v2.1	£119
Excell v2.1	£287
Multiplan v4.01	£119
Windows Draw v1.01	£115
Mouse Bus/Serial + Softw	£105
LOTUS DEVELOPMENT	
Freelance Plus v2.01	£269
Freelance Maps	£95
Freelance Plus v3.01	£269
Lotus 123 Rel 2.2	£269
Lotus 123 Rel 3.0	£319
Lotus Symphony v2.0	£353
Agenda v1.01	£221
Magellan	£92

SCO XENIX	286	386
SCO Xenix Op Sys	£357	£422
SCO Xenix Dev Sys	£357	£455
SCO Xenix Text Proc	£126	
SCO Xenix Comp SyS	£775	£905
SCO Unix (2 User) Op SyS	£341	
SCO Unix Devel Sys	£568	
SCO CGI Graph Run/Sys	£191	
SCO Multiview	£256	£321
SCO VP/ix		£321
SCO Foxbase+	£516	£647
SCO Foxbase+ Run/Sys	£191	£191
SCO Masterplan	£516	
M/Soft Word	£386	
Multiplan	£321	
SCO Eroff Documentor	£451	£516
SCO Lyrix v6.0	£386	£386
SCO Professional	£386	£386
SCO Image Builder	£386	
M/soft BASIC compiler	£451	
M/soft BASIC interp.		£256
M/soft FORTRAN comp.	£451	
M/soft PASCAL comp.	£451	
SCO XENIX-NET Server	£386	£451
SCO XENIX-NET 3 Server	£841	£1036
SCO TCP/IP (Req streams) Run/Sys	£227	
SCO TCP/IP (Req streams) Dev/Sys	£146	

Microport UNIX	286	386
Complete System	£395	£495
Comp. Sys. Unlimited	£538	£718
Runtime System (2 user)	£159	£189
Run. Sys. Unlimited	£299	£358
Unlimited User Upgrade	£159	£189
Software Development	£189	£329
Text Preperation	£129	£129
Network Services Extension (Includes Streams & RFS)		£119
DOS Merge Limited	£165	£249
DOS Merge Unlimited (Requires relevant Runtime Sys)	£165	£319
Informix		
4GL Devel	£897	£1497
4GL Runtimes	£450	£750
SQL Devel	£717	£837
SQL Runtime	£357	£417
ESQL/C Devel	£537	£657
ESQL/C Runtime	£270	£330
C-ISAM	£199	£229
Individual Manuals Each.	£60	£60
User's Guide	£60	
Advanced Programming Guide	£60	
Microport Sys V Made Easy	£25	
INTERACTIVE UNIX 386/ix		
Application Starter (1-2 Users)	£485	
Application Extender (1-2 Users)	£645	
Application Developer (1-2 Users)	£925	
Network Starter (1-2 Users)	£365	
Network Extender (1-2 Users)	£585	
Networker Developer (1-2 Users)	£965	
Workstation Starter (1-2 Users)	£325	
Workstation Extender (1-2 Users)	£715	
Workstation Developer (1-2 Users)	£1285	
Software Developement System	£485	
NFS with TCP/IP (1-2 Users)	£435	
X-Windows (Runtime)	£225	
X-Windows (X11.3 Development)	£560	
VP/IX (1-2 Users)	£280	

UNIX Servers for ALL Budgets

Starter KIT: £1195	Corporate KIT: £1895
286/12 w 1.5M RAM, 1.2M floppy, 101 keyboard, 3 I/O ports, 20M HD, mono screen. + loaded with Microport UNIX and DOS.	386/16 w 2M RAM, 1.2M floppy, 101 keyboard, 3 I/O ports 40M HD, mono screen. + loaded with Interative UNIX, VP/IX, TEN/Plus and DOS.

TCP/IP

ETHERNET

XENIX NET

NOVELL

INTERACTIVE 386 IX

MOTIF

OS/2 | UNIX

TOKEN RING

PCNX £129.00 | DOS

ESIX/V 386

MINIX £89.00 | AIX

SCO UNIX

INTEL 386 IX

Open Desktop

Centre Point Bargains - *1 minute from Tottenham Court Road tube.*

These items £50 EACH

Essential B Tree - C Doc - PC Expert - M2C2Basic - CPM86 Toolbox - ZView - OS/2 Standard Edition Ver 1.0 - Screenstar - Canon Photocopier (Used).

These items a little more BUT still a BARGAIN!

Raima dbFile (source)	£275
Raima dbRetrieve (source)	£275
CB Tree	£95
Essential Graphics (source)	£295
C Utility Library	£75
Panel Plus (source editor)	**£750**
Displaywrite 4/2	£75
Terminal Control (source)	£595
Vitamin C (source)	£149

Here's the HARDWARE

PC Brand (IBM Compatible)

286/12	(NEW)	£599
386SX-16	(NEW)	£899
386/20	(NEW)	£1349
386/25	(NEW)	£1499
386/33	(NEW)	£2299

IBM Worms Drive - (EX-DEMO) 200/cart (RRP: £2395)	£1495
IBM Quietwriter III (EX-DEMO)	£395
Qume Scripten (EX-DEMO) Postscript Printer	£2650
Compaq 386/20 Portable (EX-DEMO) 6M RAM, 100M HD	£4950

Don't forget these !!

Complete Systems.

SCO XENIX 286 £775
SCO XENIX 386 £905

UNIX

MICROPORT SYS V 286 £395
MICROPORT SYS V 386 £495
Complete Systems.

MKS Toolkit (Unix Like) £149.95

System IX Networks Ltd
55 Bedford Court Mansions,
Bedford Avenue,
London WC1B 3AD
Fax: 01 - 255 1038

Orders/Sales - (6 lines)

01-636 8210

Customer Services: 01 - 631 4818

Letters

We welcome opinions on any subject that is relevant to software development, especially feedback from articles published in .EXE. Please write to The Editor at 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked Not For Publication it will be considered for inclusion on this page.

Dear Sir,

When I first bought an MS-DOS Lattice C compiler (V3.0), Dr Dobb's magazine said that it was the hottest thing on the market and that the documentation was a tremendous improvement. With this in mind, I was surprised and dismayed, upon opening the package, that there were just two manuals, one for V2.XX and a Technical Bulletin for V3.0. There was no clue how to use the compiler. Since, at that time, I was a C novice, it took me several months trying to figure out what was going on.

History has repeated itself. Lattice V6.0's documentation also omits to tell you how to use the compiler's professed power. But that's the least of it. For example: when I tried to use the LSE installation program, LSEINST, my machine locked up. It locked up with V3.4 in just the same way, so no progress here. Next step - build the graphics library. The compiler ran out of memory (my fault, I only have 640 KB of MS-DOS memory). After stripping out *all* my drivers and TSRs, the compiler would take the 26 KB file. At least the demo worked this time; V3.4 would get to the area-fill part, then lock up.

Next stop, I played around with the dBC library (which provides access to dBASE III files). Lattice states that care should be taken when using the dBC functions, due to the interrelationships between the functions. The snag is, Lattice doesn't tell you what those relationships are! Is this an attempt to force you to buy the source to the dBC library? The Curses demo compiled without trouble, but (you guessed it) locked up my machine when run. After setting my VGA adapter to CGA emulation, the demo worked. A CGA-only library - just the thing for the 1990s. When Lattice V6.0 first arrived, I intended to try to convert my libraries and give the new software a fair shot.

I have found the compiler good only as a toy. It is certainly not worthy of the professional status that it once had. The advertisement says, 'Lattice was back on top'. What their prospective customers and registered users would like to know is: what is it on top of?

Paul S Powenski
Chorleywood
Herts

PS: Please could you tell me if there are any active clubs devoted to C/PC programming on an advanced level?

Robert Schifreen replies:

The C Users Group may be contacted by writing to the Membership Secretary, CUG (UK), 64 Southfield Road, Oxford OX4 1PA; or calling Martin Houston on 021 454 3448.

Dear .EXE,

Regarding your recent review of Lattice C, one area that always seems to be missed by many reviewers is the development cycle for applications outside Windows and OS/2. Most reviews miss out the huge amount of development that is going on control systems. Just BT as an organisation is large enough to make anyone stop and think about the scale of development in this area.

The problem is this. If you are developing a control system based on an Intel 86 family of CPUs, then the top three choices open to you are Lattice, Intel or Microsoft C compilers. Among these, Lattice is the only compiler that produces Intel compatible object files, allowing you to test your code on a PC using standard debugging tools and then use Intel's linker to produce the run-time code for your target hardware. Try doing this with either Intel C V4.0 or the Microsoft compiler. Both will fail in one way or another.

Some people still use MS C on the PC, test it and then think it will run correctly if they re-compile it with Intel C and then link it using Intel's linker. The truth is that Intel and MS C have their own unique quirks (OK bugs). For example, Intel V4.0 doesn't have real large mode. Their large mode pointers wrap around on a 64 KB boundary. But the Microsoft compiler does not suffer from this.

On the other hand, MS C doesn't even generate Intel compatible object files, making it impossible to have a direct development cycle between your PC and the target hardware.

Although Intel has made some effort in building a development cycle covering PCs, in my opinion, it still has a long way to go. It is worth mentioning that by setting the Lattice compile switches and a little thought, you can even generate code for Intel's RMX environment.

Farjad Farid
Advanced Microprocessor Development
London W8

Dear Robert,

There is a messy, but reliable, software-only solution to Mr Pote's problem of inserting data into the keyboard buffer on a PC at a lower level than Int 9h (Letters, .EXE, February 1990). Set the single step flag, and single step through the keyboard handler (whoever supplies it). When your single step handler sees that an IN AL,60h instruction has just been done, rewrite AL with the scan code you want, and resume normal execution.

Beware that the input can be in four codings which each need trapping; loading AL, loading AX, with immediate address, or via DX. Also, do not generate INT 9 - set up the stack and do an IRET to invoke the INT 9 handler.

On every PC-alike I've seen, the keyboard receiver is at port 60h.

If Mr Pote sends me a blank DOS floppy and prepaid mailer, I'll be glad to send him, by return, an assembler program which does this.

Erich Wagner
Control Telemetry
11 Canfield Place
London NW6 3BT

Dear Sir,

A recent letter in .EXE discussed ways of identifying a particular PC by taking a fingerprint. This was for use in a copy protection system. Recently, I needed to install a program throughout my company, and it was essential that users be prevented from DISKCOPYing the software and being able to run it form home. To fingerprint the machines the machines at the office, I simply looked at the time and date stamps on the hidden IBMDOS.COM and IBMBIO.COM files, which are the MS-DOS system files. These files will not change unless DOS is re-installed, and it's quicker to look up these values than it is to read the bad track table from a hard disk.

James Baron
London
.EXE

Adding to WordPerfect

If you have WordPerfect users to support, there's an easy way to write add-in programs to make life easy for them. Robert Schifreen shows how.

If you're involved in a support role, you probably have to look after users of WordPerfect. If that's the case, you may be interested in this officially-documented way of writing add-in utilities for WordPerfect version 5.0 and above. Once your add-in routine is installed, WordPerfect calls it before checking the keyboard buffer, and after every key press entered by the user. When the user presses a key, the value of this key is passed to your add-in routine, which can alter the key code if it wishes.

Full details of writing add-ins for WordPerfect are, I'm told, contained in the WordPerfect Developer's Toolkit. This costs £28 from WordPerfect Corporation. I have yet to receive a copy of this, so I wrote the WPADDIN program myself, based on a snippet of information I came across in a file on the Cix system.

When WordPerfect loads, it calls MS-DOS interrupt 1Ah, with AX=3601h. An add-in must hook this function call, and pass back to WP an address in DS:SI. WordPerfect validates the address it receives by checking that the 7 bytes preceding the address passed in DS:SI contain the string WPCORP followed by a binary 0. Assuming that the WPCORP string is found, WordPerfect will now call the routine whose address was passed in DS:SI, before checking for keyboard input, and again after every key entered by the user.

Entry Values

When WordPerfect calls the routine, it uses two registers to pass information to the routine. AX contains the code for the key just pressed (or 0, if WordPerfect is simply checking for keyboard input), and BX contains the WordPerfect state flag. This is the same flag that can be interrogated from WordPerfect macros by use of the STATE function.

From experimentation, the character code for displayable characters appears to consist of an ASCII code in AL, and a zero in AH. For non-printing characters, the value of AH appears to be 80h.

The structure of the state flag for WordPerfect 5.0 is as follows:

Bit	*Meaning, if set*
0	User is in the current document
1	User is in the current document
2	User is in the main editing screen
3	Editing, in somewhere other than main document
4	Macro definition is active
5	Macro execution active
6	Merge active
7	Block active
8	Typeover active
9	Reveal Codes active
10	Yes/No question active
11	In a list (WP5.1 only)

Upon Exit

On exit from the routine, WordPerfect expects to receive information from the add-in in AX and BX. AX should contain the key code (or 0, if there is none), which may have been altered by your routine. BX must contain zero.

Alternatively, you can return a string of characters, by setting AX to zero, and setting BX to the segment address of the start of the string. If you do this, then WordPerfect expects to find a word value (not a byte value) of 2 at BX:0, and a zero-terminated string of character codes (these are all words, too) starting at BX:2.

The value of 2 that is in BX:0 is the offset of the character about to be stuffed. WordPerfect automatically increments this as it stuffs each character, and your add-in will not be called again until all the characters have been stuffed. Once all the string has been stuffed, the word at BX:0 will point to the end of your string, so your string will not be stuffed again until you reset the word to a value of 2.

WPADDIN.COM

The program below, WPADDIN.ASM, is the source code for a simple WordPerfect add-in. You'll need MASM, or another MS-DOS assembler, to create WPADDIN.COM. Make sure you delete WPADDIN.EXE after the EXE2BIN step, otherwise you'll end up with two executable files.

WPADDIN is a simple key code detector, that makes the 'grey minus' key on an AT keyboard act like another help key. There's no reason why your add-in should be as simple as a key trapper, though - it should be possible to construct custom menu systems, mainframe communications facilities and so on.

Installation

To install WPADDIN, run it from the DOS command line. The program will then remain resident in memory until you re-boot. With WPADDIN installed, load up WordPerfect and press the 'grey minus' key. Assuming that you have typed in the listing correctly, the help screen will appear.

Incidentally, if you want to produce a full list of WordPerfect key codes, the easiest way is to add a subroutine to your add-in that prints (to the printer) any non-zero value of AX. Then just load up Word Perfect and press a key, and you'll see its key code appear on the printer. A subroutine called PRINTREGS, that prints the 8086 register set to LPT1 when called, appeared in the Code Page in the May 1989 issue of .EXE Magazine (it's also on the.EXE Disk volume 3).

.EXE

Listing of WPADDIN.ASM

```
; The demonstration code below turns the 'grey minus' ; key under
;   WordPerfect   5.x   into   another   Help   key   (F3),   by
; watching for key codes of AX=805Ch, and changing them to 8022h.

; Version 1.0.  March 1990.  By Robert Schifreen.

        code    segment
        assume cs:code,ds:code,es:nothing
        org     0100h

start:
       jmp      init      ; Jump to the install routine
intent:
                          ; All int 1Ah calls come here from
                          ; now on. Watch for Int 1Ah
                          ;(ie, our code) being called
                          ; with AX=3601h,
                          ; this is how WordPerfect asks us
                          ; for the add-in's address.
        cmp     ax,3601h        ; Is AX=3601h?
        jne     intxeq          ; If not, do nothing else.
                                ; AX was 3601h. WP is requesting,
                                ; in DS:SI, the address of our handler.
        push    cs
        pop     ds                ; Set DS to the current code segment
        mov     si,offset cs:handler ;Set SI to the address of the code
        iret                      ; And that's all we need to do.

header  db   'WPCORP',0           ; signature that must precede add-in

handler: .                        ; Now, the code for our add-in.

       cmp     ax,805Ch         ; Has user pressed 'grey minus' key?
       jne     end_handler      ; If not, take no further action
       mov     ax,8022h         ; If yes, change the key code to 'F3'.

end_handler:

        mov     bx,0             ; End of add-in handler.  Set BX to 0
                                 ; cos we haven't set up anything.
        retf            ; And exit.  Make sure your assembler
                        ; generates a real RETF here, and not
                        ; just a RET (some early versions of
                        ; MASM had a problem).

intxeq:

        db      0EAH    ; Absolute JMP (segment + offset)
intip   dw      0       ; Filled by init code
intcs   dw      0       ; Filled by init code

init:                   ; Install our Int 1Ah handler

        push    es
        push    bx

        mov     ax,cs
        mov     ds,ax
        mov     ah,35h      ; Get old interrupt vector
        mov     al,1Ah
        int     21h
        mov     intcs,es    ; Save in long JMP
        mov     intip,bx

        mov     ah,25h      ; Set new interrupt vector
        mov     al,1Ah
        mov     dx,offset intent
        int     21h

        pop     bx
        pop     es

        mov     ax,3100h                ; Prepare to terminate
                                        ; & stay resident
        mov     dx,(init-start)/16+17 ; Number of resident paragraphs
        int     21h                     ; Done

code    ends
        end     start                    ; End of file.
```

Books

Dvorak's Guide to PC Telecommunications

John Dvorak is a columnist for a number of US computer magazines. No doubt he gets lots of calls and letters from readers seeking advice, and many of those callers want to know how to get started in the world of modems, bulletin boards and communications in general. This book appears to be an attempt to answer the most common questions posed by such callers. There's no way that you can describe it as any form of technical reference (even if it does tell you how to wire up an RS-232 lead). Instead, it's a very thorough introduction to the benefits of telecommunications.

As if to show just how much effort has been spent on the production of the 1050 pages, the list of acknowledgements takes five full pages, and lists everyone who did anything for the book.

One of the first subjects discussed is the relative benefits of a number of communications software packages. The authors think very highly of a shareware package called Telix, a copy of which is conveniently enclosed on two floppy disks that are tucked into the back cover. There's also a directory of modem suppliers. Both of these listings, and many other topics besides, suffer from the fact that this book is written for the US market. However, there's still much worth reading.

Having gone to great lengths to explain how to buy a modem and link it to your PC, the book then suggests some systems that you may like to call. It covers a number of commercial (ie, expensive) specialist databases, as well as listings of dozens of free bulletin board systems.

If you or your company is considering setting up a bulletin board, you'll find a lot of advice here. A number of chapters explain how to choose and acquire the BBS host software (most of the best packages are shareware), and how to make your board friendly and, if required, secure. There's also a very brief description of the MNP error correction protocol, up to class 9, and comparisons of file transfer protocols and archiving/compression utilities such as ARC and ZIP.

Even if you tear out the 100 pages that list the US data network numbers for every state, there's still a lot of useful information left. As I've already stated, this Guide is far from being a technical reference. Instead, it lightly touches the surface of just about every facet of telecommunications that you're likely to want to know about. In that, it does a fair job. The disks, which include a copy of Telix and a number of other useful public domain utilities, only add to the value.

Author: John Dvorak and Nick Amis
Publisher: Osborne/McGraw-Hill
Price: £38.65 inc disks
ISBN: 0-07-881551-7
Pages: 1050

Inside the Model 80

Nowadays, publishers tend to include words such as 'Advanced' in the title of even the most low-level reference books. Presumably, this is done to prevent business people from feeling embarrassed when caught reading titles like 'A complete idiot's guide to Lotus 1-2-3' on the Underground.

Knowing how publishers think, I had my suspicions from the start, that this book would either be a) a very simple introduction to the IBM PS/2 Model 80, with no technical information at all, or b) yet another listing of the standard MS-DOS and OS/2 function calls, which so many publishers have used to produce whole books out of what, by rights, should be short magazine articles.

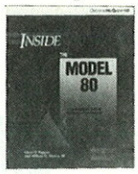
On the whole, my initial suspicions were right. There is some technical information on the Model 80, but not much. As for the DOS and OS/2 reference, there's a full 150 pages of the stuff.

Anyway, the book starts with a physical discussion of the Model 80, and the steps that led up to its birth. This description, though badly written (how many Model 80 owners, who buy technical reference books, don't know where the parallel port is?), gives some useful information regarding the internal layout of the machine. There's also a diagram of what goes where on the motherboard, and notes on how easy it is to take the machine apart (those of you who don't have a quarter will have to use a penny instead).

The Model 80 was the first machine to ship with DOS 3.30 as standard, and there's a chapter devoted to this version of DOS. It covers all the new commands, and facilities such as code pages and National Language Support. There are also details of which commands have been expanded.

By now, we're up to page 35. Skipping the DOS and OS/2 listings takes us to page 150. Here, things get quite technical. There's a discussion of the CRT controller, and the parallel and serial ports. Each of the three sections lists the registers you need to know about, and documents their bitmaps. Next comes a short, though useful, introduction to assembly language programming on the 386. The example programs cover simple tasks, such as displaying characters on the screen, and making sounds. Both DOS and OS/2 protected mode are covered, though I find it amazing that anyone would write OS/2 programs in assembly language. Assuming you'd rather stick to C, there are some sample C programs too.

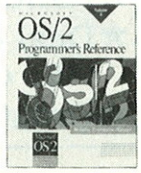
It's hard to recommend this book. As a DOS and OS/2 primer, it's a fair attempt. As a short tutorial on assembly language, again it's a fair attempt, though there are plenty of books that do it better. As a dedicated guide to the specifics of the Model 80, or even the PS/2 family in general, it fails dismally. IBM's technical reference manuals are many times better, and should be affordable to anyone who can get together the cash to buy a Model 80 in the first place.



Author: Chris Pappas/William Murray III
Publisher: Osborne/McGraw-Hill
Price: £22.95
ISBN: 0-07-881311-5
Pages: 425

OS/2 Programmer's Reference Vol 4

This is the latest volume of what Microsoft refers to as the definitive set of books for learning OS/2 programming. Around 80% of the book contains a list of function calls, with details and examples of their use. The remaining 20% is general information, from a technical point of view, on OS/2. This volume covers version 1.2 of the operating system, and includes details on both character and PM functions. Presumably, Volume 5 will cover OS/2 2.0, but we have yet to receive official confirmation from Microsoft Press.



Author: Microsoft Corporation.
Publisher: Microsoft Press
Price: £18.95
ISBN: 1-55615-259-0
Pages: 410

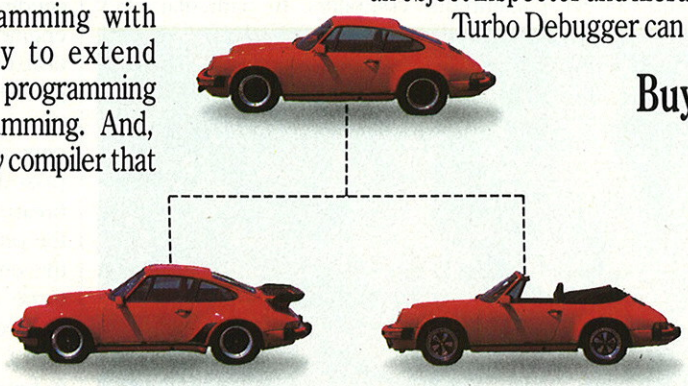
BORLAND INTRODUCES TURBO PASCAL 5.5 WITH OBJECTS

Be Objective.

Turbo Pascal,® the world-standard Pascal compiler, adds Object-oriented Programming with our new version 5.5. We combined the simplicity of Apple's Object Pascal language with the power and efficiency of C++ to create Turbo Pascal 5.5, the object-oriented programming language for the rest of us.

It's easy to extend yourself

If you're already programming with Turbo Pascal, it's easy to extend yourself from structured programming to object-oriented programming. And, Turbo Pascal 5.5 is the *only* compiler that is 100% source-code compatible with your existing Turbo Pascal 4.0 and 5.0 programs.



Inheritance provides powerful modeling capabilities by allowing objects to inherit attributes from other objects.

A fast object lesson

Object-oriented application programs more closely model the way you think. Objects contain both data and code. As in a spreadsheet cell, the value and the formula are together. Objects can *inherit* properties from other objects. For example, a Porsche Carrera inherits most attributes from base model 911, but it also sports a whale tail.

Turbo Pascal 5.5's object-oriented extensions give you code that's easier to change, extend, and support.

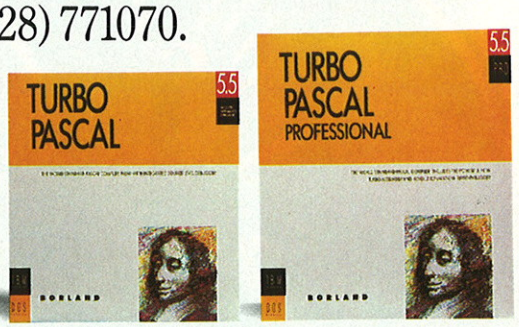
Turbo Pascal 5.5 Professional with Turbo Debugger® and Turbo Assembler®

The award-winning Turbo Debugger now includes an object inspector and hierarchy browser. And Turbo Debugger can debug any size program.

Buy objectively

Turbo Pascal 5.5 costs £99.95 plus VAT and Turbo Pascal 5.5 Professional, including Turbo Debugger and Turbo Assembler costs £199.95 plus VAT.

For further information, simply talk to your dealer or call our PRODUCT INFOLINE for a copy of our Turbo Pascal 5.5 brochure on (0628) 771070.



Turbo Pascal 5.5 Features

■ Inheritance ■ Static & dynamic objects ■ Constructors & Destructors ■ Object constants ■ Compiles @ > 34,000 lines/minute ■ New integrated environment tutorial ■ Hypertext Help with copy and paste ■ Enhanced smart linker & overlay manager ■ Support for 8087/80287/80387 ■ Integrated source-level debugging

BORLAND

Borland International (UK) Ltd.,
8 Pavilions, Ruscombe Business Park
Twyford, Berkshire RG10 9NN

CIRCLE NO. 836

Creature comforts

Jensen & Partners International has produced a brother compiler for its famous Modula-2. Will Watts has been playing with the most recent C compiler to enter the MS-DOS arena.

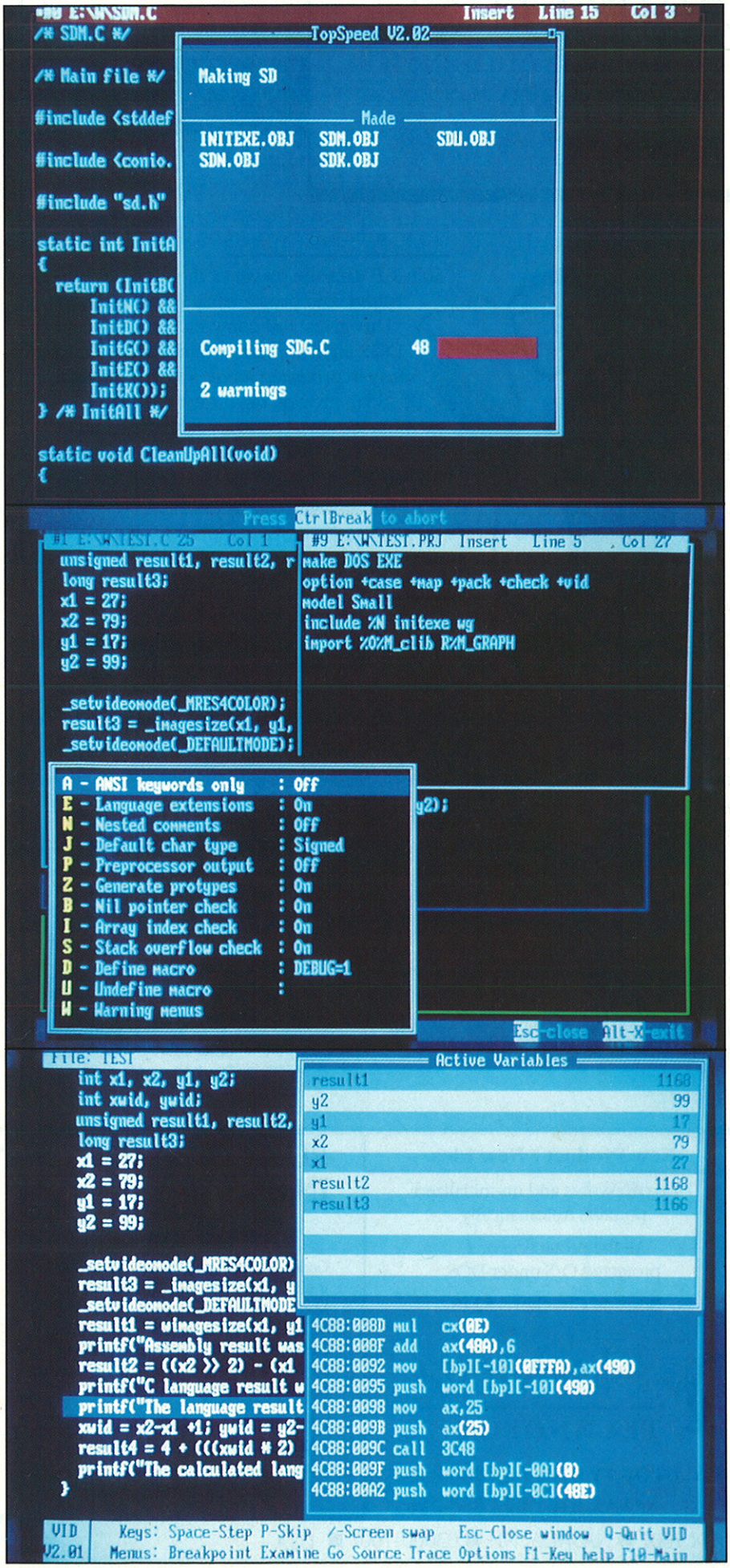
First, some pedigree. Nils Jensen, now the President of Jensen and Partners International, was, in 1982, a co-founder of a modest Irish software house called Borland. Jensen participated on such projects as the original Turbo Pascal, and SideKick, the King of TSRs, and saw the company achieve international success and acclaim. He moved with Borland to the US, and, in 1986, supervised a venture to develop a range of new in-house compilers: Modula-2, C and Ada. These products were to offer yet higher rates of compilation than the existing Pascal product, while improving and maintaining the standard of object code. To achieve this in a feasible time-scale, Jensen's team borrowed design ideas from mainframe compilers. In particular, they devised an all-purpose back end code generator, which could be 'plugged in' to a front-end language parser. All effort spent on optimising the code generator for the Modula-2 compiler automatically carried through to the C and Ada translators.

Borland's plans changed. Instead of proceeding with the development of its own C engine, it bought in Wizard C, a conventional command line compiler, and integrated it with the Turbo Pascal editor. Worries began to surface within the company about the similarity between the proposed Modula-2 product and the bread-winning Turbo Pascal. How could the products be marketed side by side? In the end, Jensen bought the rights to the three compilers (for what he coyly described as a 'not insignificant seven figure sum'), and took off, with his design team, to form JPI.

TopSpeed Modula-2, the company's first product, attracted good reviews, followed in due course by a loyal following. But Modula-2 is, perhaps unjustly, a minority-interest language, and JPI knew that if it was to make it to the big time, it would have to complete the C product. JPI finally got TopSpeed C out the door at the end of 1989. This article covers version 1.02A, which was released in mid-February.

TopSpeed C comes in three editions: Standard (MS-DOS compiler/environment, libraries, source code debugger), Extended (as above with library source, Windows support, MS-DOS DLLs, integrated assembler and various utilities) and OS/2 (as Extended Edition, plus full OS/2 support, including PM). I have been experimenting with the Extended Edition.

I chose to try the 3.5" disk version, and received six disks; four associated with the



Standard Edition, the remainder containing Extended Edition material. As usual, an INSTALL.EXE program copied the package onto my hard disk, giving me a chance to alter the default directory name setup and conserve space by rejecting the libraries of the less useful memory models. The program included a check to determine that I had sufficient space. All the files are stored in compressed format on the distribution disks, incidentally, so you would be hard pressed to avoid using this program - not that there is any good reason why you should.

Environment

A small preamble here. I have recently been on QA's excellent C++ course. To ensure that precious laboratory time is not wasted, while students wrestle with unfamiliar text editors, QA suggests that you bring the one that you are used to. With a view to combining product reviewing with education, and to consolidate my position as class poser, I took along TopSpeed.

I was very much taken by the TopSpeed Environment. Even using the TopSpeed environment for the wrong language (C++), with the wrong compiler (Glockenspiel C++/Microsoft C), I was still moving round the edit/compile/debug cycle effortlessly. A Brief user to my left was typing away at Exercise 3 at 6.30pm, when I headed off home. He was still there when I got in late the next morning. Had he been home? I couldn't be sure. He seemed to spend all of his time typing in file names: invoking Brief, invoking the compiler, getting the path name to his include file wrong, wanting to look at the solution to the last exercise and being unable to find it in the directory structure.

Before the Brief fan club takes out a contract on me, I would point out that I am not claiming that Brief does not offer the features that I am about to describe. My point is that, after about three hours practice, I was using TopSpeed more effectively than my fellow student was using Brief by the end of the four day course. Anyway, down to some specifics.

The TopSpeed Environment provides eight general purpose edit buffers (numbered 1-8) and two special purpose ones (0 and

Figure 1 - Compiling from the TopSpeed Environment

Figure 2 - The multi-buffer editor

Figure 3 - The VID debugger

FINALLY. A debugging tool tough enough to handle the DOS Nasties.

New Version 2.0

Soft-ICE™

Nasty over-write? No sweat!

Soft-ICE memory range break points help you track down memory over-write problems whether you are doing the over-writing or another program is over-writing you.

Hung program? No problem!

When the system hangs, you now have hope. With Soft-ICE you can break out of hung programs no matter how bad the system has been trashed. And with Soft-ICE's back trace ranges you can re-play the instructions that led up to the crash.

Program too large? Not with Soft-ICE!

Soft-ICE runs entirely in extended memory. This means you can debug even the largest DOS programs. And since your program runs at the same address whether Soft-ICE is loaded or not you can find those subtle bugs that change when the starting address of your code changes.

System debugging? Soft-ICE is a natural!

Soft-ICE is ideal for full source level debugging of TSRs, interrupt service routines, self booting programs, DOS loadable device drivers, real-time kernels, non-DOS O/Ss and ROMs. Soft-ICE can even debug within DOS & BIOS.

How Soft-ICE Works

Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses the 80386 to provide real-time break points on memory locations, memory ranges, execution, I/O ports, hardware & software interrupts. With Soft-ICE you get all the speed and power of a hardware-assisted debugger at a software price.

Don't want to switch debuggers?

You don't have to!

Soft-ICE can run stand-alone or it can add its powerful break points to the debugger you already use. Use your favorite debugger until you require Soft-ICE. Simply pop up the Soft-ICE window to set powerful real-time break points. When a break point is reached, your debugger will be activated automatically.

MagicCV with Soft-ICE

Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

"These may be the only two products I've seen in the last two or three years that exceeded my wildest expectations for power, compatibility and ease-of-use."

—Paul Mace
Paul Mace Software

Soft-ICE	$386
MagicCV	$199
MagicCV for Windows	$199
Buy Soft-ICE & MagicCV(W)	—Save $86.
Buy MagicCV and MagicCVW	—Save $100.
Buy All 3	—Save $186.

30 day money-back guarantee
Visa, MasterCard and AmEx accepted

1988 BYTE Award of Distinction

New Product / New Idea

Find overwrites and un-initialized pointers Automatically

All the protection of a protected O/S under DOS

Bounds Checker - $249.

Nu-Mega TECHNOLOGIES

CALL TODAY 0101 603 888-2386
or FAX 0101 603 888-2465

RUN CODEVIEW IN 8K MagicCV™

CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 to load CodeView and symbols in extended memory. This allows MagicCV to run CodeView in less than 8K of conventional memory on your 80386 PC.

NEW—Version 2.0 includes EMS 4.0 driver.
Attention Windows Developers!
Version available for CVW.

P.O. BOX 7607 ■ NASHUA, NH ■ 03060-7607 ■ USA

CIRCLE NO. 837

9). Each buffer is displayed in a removable, resizable, independently re-colourable text window. Naturally, the windows may overlap. You may open two or more windows onto the same file, for example when editing one part of a file while referencing another part, and the data will remain concurrent. If you need to look at a file, but are anxious not to change it accidentally, you can open it in read-only mode, where only repositioning keystrokes are obeyed. You can copy blocks between buffers, and cut and paste from the screen display. This naturally includes the context-sensitive help system, which pops up a screen chosen on the basis of the word currently under the cursor. As well as Environment, C and Assembly language help (which can be left on the screen, as with QuickC, so that you can continue to reference it while you work), there are some SideKick-style pop-up utilities: a table of ASCII/PC characters and attributes, a programmers' calculator (handles 32-bit arithmetic in bases 2, 10 and 16) and a simple, but handy, keyboard code utility which displays the return value, as reported by the BIOS, of incoming keystrokes.

When you quit out of TopSpeed, it remembers the names of all your open files, which buffer they were in, your cursor position, the size, position and colour of the windows and a pick list consisting of the last eight files that you accessed (but have closed), including your cursor position in those files. Each buffer can contain a file up to 500 KB in length (compare with Turbo C's 64 KB), which will be paged out to disk or stored in EMS, depending on what there is available.

TopSpeed's default editing keys are nearly all the same as Turbo C; to the uninitiated, this is Wordstar with an assortment of function and hot-keys (F2 saves the current file, Alt-C compiles it and so on). Most important control features can be accessed both by hot-keys and pop-up menus. By editing a key definition file called TSCFG.TXT, it is possible to reassign all keystrokes, including arbitrary compound key sequences such as <Home End>, and so create an emulation of your own favourite editor. (JPI would be well-advised to supply versions of TSCFG.TXT set up to imitate a handful of the most popular editors; but for once the company missed a trick.) You can also change the structure and content of menus, even adding items which invoke external MS-DOS commands. This would be an excellent place to hook in an external compiler - you can have the environment prompt the user for a filename. To run the Glockenspiel C++ compiler, however, I used the simpler device of a keyboard macro. These are recorded sequences of keystrokes which can be assigned to a playback key. Macros are stored in ASCII text files, for easy editing.

I could continue to list TopSpeed Environment features, but I might drive you back to your copy of Woman's Weekly

I could easily continue to the end of this review listing TopSpeed Environment features, but fear that I might drive you back to your copy of Woman's Weekly. So instead I will record one important limitation and give a summary impression. The limitation is: no mouse support (which is fine by me, but some people must have it, apparently). The general impression: if you know the Turbo C/Pascal Environment, then TopSpeed (unsurprisingly) shares the same philosophy of design, is just as neat and is about an order of magnitude more powerful.

The Compiler

The C compiler can be invoked in two ways. There is a conventional command line utility, for use by version control systems and the like. From the integrated environment, you hit Alt-C and, provided that you have a .C file or a .H file in the current edit buffer, it will start up the C compiler. (The extension is important because the TopSpeed Environment uses it to determine whether to start the C compiler or the assembler. The latter is used to 'compile' .A files.) As well as the usual line count, the program also displays a sideways bar graph, which indicates the proportion of the source file which has been processed. If any errors are found, the compiler places the offending file in buffer 0, with the cursor positioned at the first mistake. There is a mechanism which lets you skip from one error to the next.

As for the quality of object code, refer to Figures 4 and 5. These show the results of compiling and running the Plum Hall C benchmarks, using TopSpeed and half-a-dozen well known MS-DOS C compilers. The benchmarks were run on a 20 MHz 80386 machine. In all cases, I enabled all the compiler optimisation switches that I could find; including any 'generate code for 286/386 processor' type options. My results suggest that, while TopSpeed is a strong contender for second place, it falls short of the quality of code produced by Watcom's famous product.

	register int	auto short	auto long	int multiply	function call+ret	auto double	compile time(secs)	.EXE size
TopSpeed C V1.02A	0.26	0.26	0.83	1.42	1.27	51.6	12	27590
Microsoft C V5.1	0.33	0.32	1.60	1.10	1.10	160.0	18	31066
Watcom C V7.0	0.24	0.24	0.75	1.20	1.10	8.01	27	18119
TurboC V2.0	0.35	0.38	1.00	1.20	1.10	110.0	6	27348
QuickC V2.0	0.39	0.44	1.70	1.70	1.70	160.0	14	32146
Lattice C V6.0	0.31	0.35	0.98	1.23	1.61	27.1	258	27758
Zortech C++ V2.0	0.31	0.31	1.04	1.20	1.16	18.7	14	24160

Except for compile time and .EXE size, the results are shown in microseconds per individual operation, using a 20 MHz 386 machine.

Figure 4 - Plum Hall C Benchmarks, with compile time and file size

I did encounter one example of over-zealousness in TopSpeed's optimiser. I declared a variable `volatile unsigned long`, but the compiler chose to ignore the `volatile` keyword, so the optimiser placed a copy of my variable in two registers. Unfortunately, the real variable was being updated from an interrupt routine, so this decision had a catastrophic effect on my code. I had to disable a large number of default optimisations before the routine worked correctly.

The Plum Hall suite does not include a test for speed of parameter passing. It is a boast of both the Watcom and the TopSpeed compilers that function arguments are passed in registers whenever possible, instead of in a stack frame. This can save time, but it does mean that interfacing with other languages and compiler's libraries is potentially messy. Like Watcom, again, TopSpeed offers two systems for changing the calling convention. One is to declare a function using the keyword `cdecl`, which forces Microsoft C calling conventions, but is not ANSI; the alternative is to use the more flexible `#pragma` directives, which let you force functions into pretty well any calling convention that you care to name.

Like all modern MS-DOS compilers, TopSpeed comes with its own linker. Atypically, JPI has chosen to add some non-standard features. Borrowing an idea from C++, TopSpeed's linker offers type-safe linking. If you declare a function

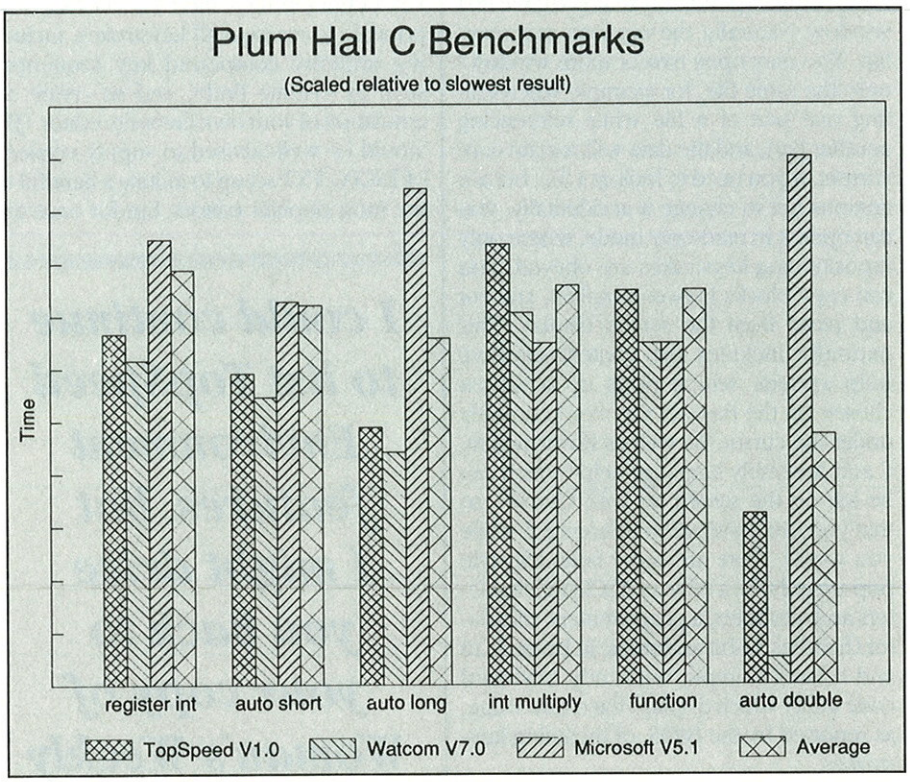
```
double fn(int i)
{
...
}
```

and attempt to call it from another module, with absent or incorrect prototype, like this

```
int i;
i = fn(2);
```

then you will get a link error. Unlike C++, the system does not depend on creating a new identifier, so there is no difficulty in linking to, say, assembly language routines. There is also a feature called 'smart linking', which lets you build a library with just one source file. The linker is able to pluck out referenced functions and data objects from the resulting library file, without including redundant code. Normally, to achieve this effect, you have to build libraries from dozens of little source files, which is no small hassle. The Extended Edition includes an MS-DOS based Dynamic Link Library capability (there is no conventional overlay system). Projects which use this can be migrated to OS/2 by rebuilding the unaltered source - provided that you have the OS/2 version of TopSpeed, of course. The Extended Edition also supports Windows development, but you will still need a copy of Microsoft's SDK.

Figure 5 - Graph of Plum Hall C Benchmarks



Make it Easy

The project make system you will really like. TopSpeed uses its own conventions for make files. This might seem like bad news, until you discover that you don't have to include any dependencies; essentially you need only specify the type of target that you intend to produce (eg DOS EXE, OS/2 EXE, DOS DLL etc), provide any link options (debugging information, map files and so on), specify the names of your source files (which can be assembly or C) and libraries, and you are through. Most of the creation process is done automatically from within the TopSpeed Environment. The make component deduces all the `#include` dependencies by reading your source code. Not only does it compare date/time stamps of source files, it stores a checksum of the values in the objects that it makes. Thus, if you retrieve a backup copy of a source file (with an old date), it knows that it must perform a remake - unlike conventional MAKEs, which force you to resort to utilities like TOUCH. The make utility is available both inside and outside the TopSpeed Environment.

One example of over-zealousness in TopSpeed's optimiser had a catastrophic effect on my code

The source level debugger, VID, was a bit of a disappointment, although I find it hard to pin-point exactly why. All debuggers, in my experience, have a tendency to crash a lot - the dubious quality of the code that you are testing ensures this - but VID seemed to over-indulge in the habit. Surprisingly, in view of the excellent design of the editor, control of VID was not particularly clean, and inconsistent with the environment. I kept hitting 'Alt-letter' to select a menu item, where the program wanted the letter key by itself. VID supports all the usual trace/watch/breakpoint features, can drive two monitors and store some of itself in EMS to maximise available DOS space. It doesn't do the voguish debugger trick of installing a small kernel on a target machine and letting you control it remotely via a serial link to another machine, but I dare say that it will come with the next release. A bonus utility called *Watch*, which is a removable TSR which lets you monitor any application's access to MS-DOS interrupt 0x21 calls, partially makes up for VID's shortcomings.

POWER AND FLEXIBILITY

BRIEF 3.0 increases your productivity for just £149.95*

*VAT extra, FREE delivery

BRIEF has established itself as the leading programme editor on the market today. Now, with the launch of the latest enhanced version – BRIEF 3.0 – it offers even better productivity.

ENORMOUS UNDO POWER

BRIEF 3.0's enhanced UNDO capability takes the fear out of programming. BRIEF 3.0 allows you to undo any command that affects the cursor or the text (including global changes) up to 300 times, even back to the beginning of your session. It's the safety net you need to program with confidence.

REAL FLEXIBILITY

There's built-in support for 36 languages including Assembler, BASIC, C, COBOL, dBase, FORTRAN and Pascal. And thanks to its complete macro language you can make BRIEF 3.0 fit your own programming style with ease: change key assignment, commands, prompts, colours – even the indenting styles.

BRIEF 3.0 – new benefits

- With our new C-like macro language, CBRIEF, C programmers can write macros on the fly and not have to switch between C and a different macro language
- The new source level debugger works with both the original BRIEF syntax and the new CBRIEF macro language
- Maximum lines per file has been increased to 4 billion
- We've added to and improved template editing and smart indenting for ADA, Cobol, Basic, Fortran, Modula-2, Pascal and C
- We've added editable, multiple keystroke macros that can be saved and restored
- You can even save and restore your window set-ups

Plus the features that made BRIEF famous

- Flexible windowing
- Unlimited number of files
- File size limited only by desk space
- Regular expression search and replace

No other editor puts this much power and flexibility in your hands. So if you're looking for extra productivity purchase your copy of BRIEF 3.0 today.

Solution Systems

SOLUTION SYSTEMS are the UK distributors for BRIEF 3.0. When you buy from us you get the authorised UK version of BRIEF 3.0 dual media (3½" and 5¼") plus bonus macro diskette.

Special phone line for Access/Visa card holders. Save time by phoning **0763 73457** between 9.00 and 6.00 Monday to Friday to place your order.

N.B. Official purchase orders from local authorities, PLCs and educational establishments are accepted by post or by Fax.
Fax no: 0763 73460

BRIEF 3.0 The Programmer's Editor

Send to: Solution Systems UK, The Red House, 84 High Street, Buntingford, Hertfordshire SG9 9AJ

Please rush me my copy of BRIEF 3.0

☐ **I enclose cheque to the value of £172.44 (£149.95+VAT) crossed and made payable to Solution Systems UK.**

Orders despatched same day.

NAME ADDRESS

POSTCODE

☐ **Please charge my Access/Visa* account the sum of £172.44 (£149.95+VAT)**

Card No **Expiry Date** **Signature**

*Delete as appropriate

EXE 4.90

CIRCLE NO. 838

The Extended Edition also includes a simple one-pass assembler, which may be operated from the command line or environment. The problem with this utility is that it has a completely non-standard syntax, derived from Modula-2. For example, comments are delimited by (* and *), and semicolons are used as statement separators. This means that converting even a small 'standard 8086' assembly language file is a non-trivial task. All the assembly directives are different, there is no support for 80286 and above instructions and there are no macros. This makes it a very restrictive tool, which is a pity, because it would be very useful to have a good assembler integrated into the environment. Incidentally, some of the source code for the libraries and miscellaneous bits and pieces, such as startup routines, are written in this assembler.

Libraries

The libraries include a complete set of ANSI functions (the compiler claims full ANSI compatibility), plus what appeared to be a complete implementation of the Microsoft C library (but I admit that I have not done a function for function check). This includes the section where nearly all vendors like to invent their own set of standards: the graphics functions. TopSpeed also implements Borland's calls for text window handling. To perform BGI graphics, however, you will need a copy of Turbo C; but JPI does include special header files to ease the strain of incorporating BGI graphics in a TopSpeed program. I ported a Turbo C graphics program to TopSpeed without any hitches.

```
module WG

segment WGRAPH(CODE, 28H)
segment GDATA(CODE, 68H)

select GDATA
(* Reserve screen buffer *)
public _hidvid:
org 4000H

select WGRAPH
(* Stack frame equates *)

X1          =   06H
Y1          =   08H
X2          =   0AH
Y2          =   0CH
HEAD_LENGTH =   6

(* Returns in ax number of bytes
   required to store
  rectangle X1,Y1,X2,Y2 *)

public _wimagesize:
    push    bp;         mov     bp,sp
    push    dx
    mov     dx,[bp][X1]
    shr     dx,1;       shr     dx,1
    mov     ax,[bp][X2]
    shr     ax,1;       shr     ax,1
    sub     ax,dx;      inc     ax
    mov     dx,[bp][Y2]
    sub     dx,[bp][Y1]
    inc     dx;         mul     dx
    add     ax, HEAD_LENGTH
    pop     dx;         pop     bp
    ret     far 0
end
```

Figure 6 - Example of TopSpeed-style assembly language

The time slicing routines, which allow you to create multi-thread MS-DOS programs, are a novel inclusion. This feature is borrowed from Modula-2, and it requires a special compiler memory model to support it. (Six standard models are supported, in effect these are Small through to Large. It is also possible to create customised variations.) The function calls work under both MS-DOS and OS/2. There is also a (second) more powerful, re-entrant library of text window routines, designed to be accessed from multi-thread programs, and a set of mouse interface routines, closely based on the assembler level API defined by Microsoft. The programs that I re-compiled using TopSpeed's libraries generally worked as before. The exception was a call to `fopen()` to try to access a file which had been made read-only. Although the access mode had been set to `"rb"`, this failed because of a bug in the library.

The time slicer, which allows you to create multi-thread MS-DOS programs, is borrowed from Modula-2

The documentation consists of five perfect-bound manuals: User's Manual, Library Reference, Language Reference, Language Tutorial and TechKit (the latter covers all the extra software supplied with the Extended Edition). The text is comprehensive and clearly written throughout. The tutorial, by K N King, is certainly good enough to be a 'real' book, and I imagine would be sufficient material for a beginner to teach himself the C language. My only complaint is that there are some signs of the material being thrown together at the last minute. For example, cross-references given in the index frequently seem to 'miss' by a couple of pages. Otherwise, this is as good a set of manuals as I have on my shelf.

Conclusion

TopSpeed C is an innovative package, displaying many original features (as my estate agent would put it). The implementation is uniformly fast and slick and, although it is perhaps not quite as good as the manufacturers would have you believe, the object code is demonstrably tighter than the other MS-DOS integrated compilers (Turbo C and QuickC). It offers Windows and OS/2 support immediately, not at an unspecified future date. The documentation is good, and the entry level price, although above that of the cheapest compilers, is competitive.

Here is the 'but' list. There are some nasty incompatibilities with existing 'de facto' standards. The absence of a conventional MAKE probably doesn't matter; if you wished to rebuild an old project, you have only to alter the compiler invocation in the MAKE file and use your original MAKE utility. The different style of assembly language is more serious. Converting between the MASM and JPI syntax standards is a pain, and I would be unwilling to attempt a serious assembly project using JPI's system, despite the convenience of the integrated environment.

The other drawback to the product is its youth. However careful and thorough JPI's programmers have been, this product is in its infancy. The code generator, editor and debugger components have been field-tested by Modula-2 users; but the compiler front end and libraries are entirely new. During my review, two reasonably serious faults showed up; I think it likely that you would find quite a few more during sustained use. But I should like to emphasise the distinction between TopSpeed and, say, Lattice's most recent release, which was onto version six and *still* seemed delicate. I expect JPI to do better.

The TopSpeed language project is an ambitious and exciting undertaking; I wish JPI the best of luck with it. There are a lot of things right with this package. In some sense, TopSpeed C is the 'original' Turbo C, and it has displaced the actual Turbo C as my compiler of choice.

.EXE

TopSpeed C costs £149.00 (Standard Edition), £295 (Extended Edition) or £370 (OS/2 Edition). All prices exclude VAT and P&P. The product is available direct from the manufacturer - call JPI on 01 253 4333.

Proven, Integrated Technology for CICS.

Every time you're about to start tackling your applications backlog, application errors and other problems can send you right back to where you started. But now you can take a giant step forward with CICS Productivity Centre.

Here, in a single source is technology that addresses all critical phases of application development and maintenance:

- Testing/De-bugging
- System Testing/Quality Assurance
- System Stability
- Application Development
- Maintenance/Operations

CICS Productivity Centre

CICS Productivity Centre saves you valuable time in your operations and because it costs far less than you would normally pay for separate tools offering these capabilities, it saves you valuable money too.

Unique Interactive Capabilities

CICS Productivity Centre was designed to provide the most comprehensive approach to increased productivity, with full integration as a primary requirement.

As a result, you get a complete set of CICS utilities:

- An interactive testing approach that lets programmers detect and correct all application abends known toCICS - before the application fails.
- The ability to eliminate manual steps and improve accuracy during all types of CICS testing.
- Unique technology that detects and actually repairs CICS damage to prevent system failures.
- Centralised message control, and automatic response to critical CICS messages.
- A symbolic analysis component for resolving CICS production system application failures on line.
- Technology that pinpoints application transactions causing system problems.
- An application development system featuring prototyping capabilities and re-entrant compiled code.
- The ability to manage CICS resources on-line.
- A comprehensive journal management system, providing forward and backward recovery of VSAM and BDAM files.

Critical CICS Needs

Here's a complete checklist of the needs that CICS users have identified as critical to maintaining high levels of productivity. CICS Productivity Centre's interactive technology addresses all these requirements. If your needs are on here then you need CICS Productivity Centre.

- ✓ Faster and more reliable application debugging and testing.
- ✓ A dramatic reduction of the time and costs associated with application development and maintenance.
- ✓ More reliable, higher quality applications and systems.
- ✓ A faster, more accurate method for performing all CICS tests.
- ✓ Faster, more efficient creation of multi-user applications.
- ✓ Reduction of the application development backlog.
- ✓ Resolution of production systems abends, from the safety and convenience of test CICS.
- ✓ Automatic repair of system damage.
- ✓ Faster resolution of production problems.
- ✓ Prevention of recurring system errors.
- ✓ Greater system availability in test and production.
- ✓ Forward and backward data recovery.
- ✓ The elimination of scheduled and unplanned CICS downtime.
- ✓ A more productive environment for programmers.

The Safe Choice

Since 1969, On–Line Software has been recognised as the leading source of CICS software, education and consulting. Now we've built the same reputation in DB2. And we've been designated as an IBM* Business Partner and Application Specialist.

To protect our customers' investment in technology, our software is offered with a guarantee.

On·Line Software
INTERNATIONAL

Please send me more information on On–Line Software products

Name ..... Company .....

Address .....

..... Postcode .....

IBM Business Partner
Authorized Application Specialist

*IBM Is a registered Trademark of International Business Machines

Complete coupon and post to: On–Line Software International, Tenterden House, Tenterden Street, Hanover Square, London W1R 9AH
Tel: 01–631 3696 Fax: 01–493 5102 Telex: 28958 MPGLON G

.EX1

CIRCLE NO. 839

I Can't Work it Either

Knowing how a user's mind works can help you write software that will be easier to use. Jeffrey Goldberg opens the psychology textbook at page 1.

You can see the trail of dispirited first timers, Marks & Spencer's Chicken Masala in hand, trying to operate the microwave. It's always the same: put the chicken in the oven, press Start, come back in five minutes. Of course, the chicken is as cold as when it left the bowels of the fridge. You see, you have to press Time, type in a time (being sure to press hard on the hammer-sensitive keyboard) and *then* press Start. Don't type in the time before pressing Time, as it will reset the time to zero. Naturally, the time entered has an unknown relation to the one in the instructions on the Chicken Masala packet. Why, oh why, does it seem difficult?

The art of providing this level of difficulty is called user-interface design. In many cases, designers think this is the easy part of the design, which they can leave until last. Hogwash. If you allow that to happen, you get a microwave oven that people buy, but cannot operate.

Think People

Software designers, when designing user interfaces, tend to make educated guesses as to how the user will behave. Before you start designing the user interface, you should know how people really *do* behave when we want to do something. There are four stages that we go through:

1. *Form an aim in our minds.*
2. *Select an action from the current situation, using a list of actions in our long term memory.*
3. *Perform the action.*
4. *Check the results of performing the action.*

People use their short term and long term memory to accomplish the task. Short term memory lasts only a couple of minutes, and is the memory that governs how we cope with display items such as menus or a dialog box. Typically, this memory can only contain about seven items at one time.

Long term memory governs how you learn to use the computer. Various theories abound about how it works, but let's take one nearest to the computer world. Imagine your memory is a compiler. A compiler writer splits the parsing of the code (the syntax) from the generation of the intermediate object (the semantics). Similarly, your brain separates the computer dependent details (syntax) from the aim of the task (semantics). For example, I want to type the letter H (semantics) so I move my finger to the 3rd row, 7th column of the keyboard and press the key (syntactic).

Supply a consistent image of the program to the user

You can see that syntactic knowledge depends on the computer system and its software. The knowledge can only be acquired by rote memorisation and repetition, so it quickly fades over time. This is the knowledge affected when IBM changed from the 84 to the 102-key keyboard, and all our fingers pressed the wrong keys. It is intolerant of minor variations in software or hardware. It is also the knowledge that novices or intermittent users find particularly difficult to retain.

You can further split semantic knowledge into the task and computer concept. The task concept is the object and actions, such as 'I want to underline this sentence'. The computer concept, on the other hand, is the knowledge to press the keys Shift-F8 and Alt-U. People may be familiar with the one concept and not the other, or with both, or with neither. For example, I have sufficient knowledge to load and run Pagemaker, load a file, and use some of the tools, but kerning reminds me of popcorn rather than a font. I have grasped the computer concept, but don't know the task concept. On the other hand, I tried WordPerfect a few days ago on the PC; I wanted a boxed paragraph but couldn't figure it out. Here, I had a clear task concept, but a fuzzy computer concept. Obviously, some novices don't know what a word processor is or how to operate it - they have neither concept.

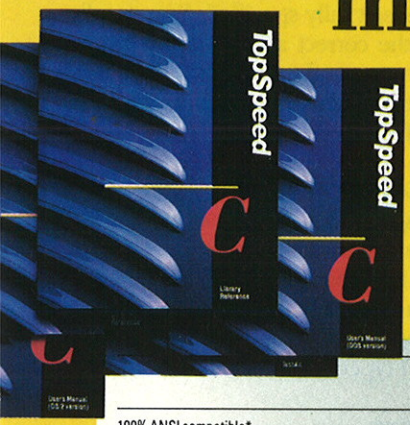
Models

People form models in their minds, of how objects work. These models help to explain the events as the person perceives them. For example, imagine a user with a word processor, saving a file to a floppy disk.

When the program says it's saving the file, the user might imagine the word processor gathering up the sheets of paper and stapling them together. Then the floppy disk light goes on. The user may imagine the Disk Genie at work. As the disk drive whirs, this could be the Genie grunting contentedly. Once the grunts stop, the Genie is asleep, and the file is saved.

This model explains all the actions of saving a file. It even copes with disk errors. It isn't near the truth, but who cares? It provides an explanation for the actions, and that's all the brain requires. Models aren't fixed, but keep on changing to keep up with each new action/reaction.

Can your compiler meet the challenge?



We invite you to take an existing program and compile it using TopSpeed C. Then, compare the overall performance with the compiler you now use. If you are not 100% satisfied, return the entire package to us, and we will refund all of your money.

	TopSpeed® C Version 1.00	Microsoft® C Version 5.1	Turbo C® Version 2.0	Watcom® C Version 7.0
100% ANSI compatible*	✓			
Integrated environment	✓		✓	
Pass parameters in registers	✓			✓
Expand any function as inline code	✓			
Supports OS/2	✓	✓		
DOS Dynamic Link Libraries *(overlay code linked at load-time)*	✓			
Smart linking *(only referenced code and data linked into .EXE)*	✓			
Type-safe linking *(function parameters and memory model checked at link-time)*	✓			
Fully automatic make works across libraries	✓			
Time-sliced scheduler for multi-tasking under DOS	✓			
Short pointers in any segment	✓			
Hypertext help with library online	✓		✓	

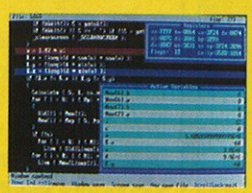
Written by Neil Martin of the British Standards Institution (BSI) and printed in Personal Computer World June 1989, page 241.

TopSpeed®

C

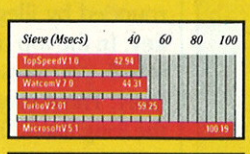
TopSpeed's seamlessly integrated multi-windowed environment.

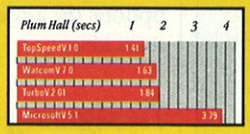
VID (Visual Interactive Debugger): a source-level, multi-windowed symbolic debugger.



To buy in the USA and Canada: Jensen & Partners International
1101 San Antonia Road.
Suite 301 Mountain View
CA 94043
Phone: 415 967 3200

Standard Edition	$199
Extended Edition	$395
OS/2 Edition	$495





Standard Edition—
Optimizing DOS compiler, integrated source-level debugger, smart linker, automatic make, complete ANSI library, 6 memory models, BGI Interface, time-sliced scheduler, MS-DOS/BIOS interface, superset of MS and Turbo C libraries, and more.

Extended Edition—
Standard Edition plus full source code to libraries, post-mortem debugging, MS Windows support, support for DOS DLLs (overlays), assembler, disassembler, profiler, DOS call monitor, and more.

OS/2 Edition—
Extended Edition (except for DOS compiler) plus ability to generate DOS executables, full support of OS/2 Kernel & PM calls, OS/2 smart linker, fully automatic generation of DLLs, source for protected mode libraries, and more.

Benchmarks measured by Mark Hamilton, November 24, 1989. Copyright © 1989, PC Business World and CW Communications Ltd.

TopSpeed C:

Standard Edition £149.00
(DOS compiler and VID)
Extended Edition £295.00
OS/2 Edition £370.00

All prices are excluding VAT, postage and packing.

64 page TopSpeed C Technical Specification available on request.

Phone for TopSpeed Modula-2 compiler and toolkits technical information and prices.

To Order:

Phone: 01 253 4333
Fax: 01 251 1442

Phone for full details.

Credit cards Visa/MC Access

Jensen & Partners UK Ltd
Unit 21
63, Clerkenwell Road
London EC1M 5NP

30 days unconditional money back guarantee
30-day unconditional money-back guarantee.

1101 San Antonio Road, Ste. 301
Mountain View, CA 94043
Phone: (415) 967-3200
Fax: (415) 967-3288

CIRCLE NO. 840

So, what your software should supply is a framework to allow people to build fairly simple models. You must, therefore, supply a consistent image of the program to the user.

Visibility and feedback are the basis for the models. You should make every function visible, so that someone can immediately tell the state of the system and decide on a course of action. Once decided on the course of action, the person looks to see what happened, so it is very important to make sure that you give give lots of feedback.

Errors

Everyone makes mistakes, slips, misunderstandings. Obviously your program should cope with these mistakes, so let's look at what these mistakes typically are:

- *Capture Errors*

A frequently performed action suddenly takes over from the one intended. For example, you are typing into your word processor. Suddenly, you realise that you need to get some milk for the coffee. You turn off the computer and walk down to the shops. Thinking about getting the milk captures the action of saving the file, so you lose all your work.

- *Description Errors*

The intended action is like others, but the sequence isn't fully specified. You usually perform the correct action on the wrong object. For example, you try to exit Borland's Turbo Debugger (ALT-X) with the WordPerfect exit key (F7).

- *Data Driven Errors*

Something on the screen, or outside the computer, intrudes into your train of thought. For example, I recently typed in the filename of this article, while listening to the pop group Texas. Instead of typing UI.DOC, I typed TEXAS.DOC instead.

- *Forgetting*

You forget what you were doing. This often occurs during or around a cup of tea. For example, you type COPY on the command line, get a cup of tea, come back and forget why you typed COPY.

Unshifted Function Keys

F1	Help.
F2	Save File.
F3	Load File.
F4	Print.
F5	Zoom. Toggle between making the window full and normal size.
F6	Switch Windows in a multi-window system, or an alternative to F8 for block marking.
F7	In a WordStar-based system, Mark the beginning of the block. Otherwise unspecified. One approved use is moving one record towards the start of the file in database systems.
F8	In a WordStar-based system, Mark the end of the block. Otherwise, enter block marking mode. In systems without block marking, such as databases, you can use F8 to move one record towards the end of the file.
F9	In a twin menu system, open the alternative menu.
F10	Enter the main menu.
F11	Unspecified.
F12	Unspecified.

The Cursor Keypad

The cursor key pad causes many problems, because programmers haven't stuck to the original IBM recommendations. My suggestions are:

Arrow keys	Move the cursor.
Home	Move to the beginning of the line.
End	Move to the end of the line.
PgUp	Moves the cursor a distance of one screen (minus two lines) towards the start of the text.
PgDn	Moves the cursor a distance of one screen (minus two lines) towards the end of the text.
Ctrl-Home	The beginning, for example the top of the letter or start of the database. A controversial decision, since Ctrl-PgUp is more logical. However, an increasing number of DOS programs use Ctrl-Home, as do Windows and OS/2.
Ctrl-End	The end, for example the bottom of the letter or end of database.
Ctrl-Left Arrow	Move left by one word.
Ctrl-Right Arrow	Move right by one word.
Shift-Arrows	Block marking from the cursor position.
Shift-Home	Block marking from cursor position to beginning of the line.
Shift-End	Block mark from cursor position to end of the line.
Ins	Toggle between insert and overwrite modes.
Del	Delete the character to the right.
Grey Plus	Increase value, for example getting deeper and deeper into an outline, or changing the format of a list.
Grey Minus	Decrease value.
Shift-Grey Plus	Increase to maximum value.
Shift-Grey Minus	Decrease to minimum value.

Figure 1 - Suggested keyboard layout

Learning From Errors

A man once told me that you learn more from watching other people make mistakes than watching them get it right. Let's peek into an everyday MS-DOS computer conversation. The user types `DEL \*.* <ret>` , and MS-DOS asks `Are you sure (Y/N) ?`. The user says that yes, he is sure, and presses `Y`. Later on, he realises that he was *not* sure, and out comes the copy of the Norton Utilities.

The problem here is that the 'Are you sure?' message comes immediately after the user has typed the command. At that stage, the person is still happy with the delete action, so it's a worthless prompt. Much better would be to allow the person to recover any deleted file until a certain date, or until the disk runs out of space.

A common description error which happens with increasing regularity is the lengthy lists of similar switches or menu items. The best example I've come across is a Sony TV remote control; this has 40 identically shaped and coloured keys, surrounded by illegible text obscured by the key. In the top right corner is the OFF key, which you press with monotonous regularity. These examples happen in software: the long list of file names, the mountain of options on a menu and the easily-pressed reset key.

Designing for Errors

Once you know that people are going to make errors, you are half way to preventing

PCL 4.0 instead of BASIC, PASCAL or C for DOS and OS/2

Produce blazingly fast applications in record time. PCL has 4GL-like super functions but combines them with the execution speed of the best compilers.

PCL programs are easier to write than BASIC, more readable than C and as well-structured as PASCAL.

- **Supports 4 interrupt driven comms ports, 2-19200 baud.**
- **Optional SQL interface to ORACLE.**
- **Dynamic record structures.**
- **Built-in windowing & menu system.**
- **Multi-tasking & shared variables in OS/2.**
- **Ram-resident in DOS.**
- **Background tasking in keyboard wait-state in DOS and OS/2.**
- **Array, time/date & extended scientific maths, 80x87 support.**
- **Fastest sort, search, text & file handling in the business.**
- **Run-time execution of source lines.**
- **Network support.**
- **Licence free run-time module.**

DOS	**Demo Disk**	£5
DOS	PCL 4.0	£195
DOS	Oracle/SQL tool kit	£65
OS/2	PCL2 2.0	£220

Postage included. Add 15% VAT. Specify diskette type: 3.5" or 5.25" and operating system (DOS or OS/2).

Orders for the demo diskette must be pre-paid. All others are dispatched with invoice, for payment within 30 days.

CALEND

PO Box 94, Twickenham, TW2 6DD, England
Telephone: 01 894 7409 FAX: 01 755 0670

the errors happening in the first place. If an error occurs, your program should attempt to find out what caused it. Keep notes of the causes of common errors, and add specific code to future versions of the software to prevent these errors happening again.

Most often, designers try to prevent errors by preventing the particular action. In the jargon, this is a 'forcing' function. You must be very careful with doing this, as the user usually has a good reason for doing the action, and doesn't like being stopped from doing it.

A bitter example was a SideKick Plus bug, where saving a file as CON.TXT did nasty things to the screen (because saving data to CON will write data to the standard output device, rather than to a file called CON). The development team (of which I was a part) worked around this bug by preventing the user from writing any file whose name was an MS-DOS device name. This prevented any unwanted screen corruption, but also stopped people using the SideKick printing trick of writing a file to LPT1.TXT. To this day, SideKick Plus users are still regretting that decision.

Going Backwards

You won't be able to prevent all errors with preventive code, so you should make it possible to reverse or undo every action. In the MS-DOS example, you could take Norton's Disk Doctor or PC Tools and recover the file. On the supposedly friendlier Macintosh, you would have a difficult task. Clearly, the Macintosh designers didn't understand errors. Unfortunately, if you build in an undo facility, people come to depend on it. The facility had better be reliable, or you'll get your phone bombarded with grief-stricken users.

Prevention and recovery of errors requires that the person knows they have made an error. Typically, they won't realise the error until long after correction was possible. You should make it easier to discover errors, by providing feedback as well as precise error tracking. Don't think of a person making errors; just the normal actions to complete a task. Above all, don't punish the poor dear.

Let Them Play

People want to have fun while at work but, too often, the tools they use are boring. Computers, with their natural interactivity, can make the tool into something more interesting. It's up to you to make this happen. MacPaint is a classic example of a designer making a tool fun to use. People seeing MacPaint just itch to touch the mouse and doodle. There are four things that make it fun:

- *Accessible*

You can start doodling immediately. There's no naming of files, setup screen or edit mode to worry about.

- *Obvious*

It's obvious what to do. The presentation makes it clear what to do, and matches the user's model of the program.

- *Feedback*

The person can immediately see the result of each action.

- *Inviting*

The Presentation is inviting. It looks so good that people want to use the software.

These attributes apply to an accounting program for funeral directors, just as much as to a paint program. Don't be put off by the PC's text screen - you can still use colour, and the semi-graphic characters.

Modes

One of the several hundreds of holy grails in computing is a program where you can do any action, regardless of the previous activity. This is the infamous modeless program. An example would be a word processor where you can type some text, then search the file for a string, then carry on typing text. Studies have shown that less modal (you can tell they are American studies) programs are easier to learn and easier to use. People forget whether they perform a particular action in a particular mode, or they show surprise when prevented from doing a particular action in a particular mode.

Unfortunately, real life has modes. When you are scribbling on paper, you are only using one pen. To select another pen, you go to your pencil case and take out another one. In computer terms, you exit drawing mode, go into select mode, select a new pen and return to drawing mode. Does this sound familiar? Yes, MacPaint is the least known example of a program filled with modes, but most of these modes match real life so they don't affect ease of learning. Unfortunately, what they do affect is ease of use, as anyone who has used MacPaint will testify.

Your program must not use modes as forcing functions, to prevent errors or correct anomalies in coding. For example, in Microsoft Word Version 4, if you enter more than 50 KB of text without saving, then the word SAVE starts flashing at the bottom of the screen. You are now in save mode, prevented from doing anything except saving the text, because of a limit in the program's design. If you were pasting from SideKick Plus into Word 4, this was a most annoying use of a mode.

Visibility

MacPaint makes the drawing tools and palettes visible to the left and bottom of the main screen. This makes it obvious that, to start doodling, you could just click on one of the tools. The mouse and graphics screen make MacPaint what it is. Unfortunately, most of us live in the harsher non mouse-driven PC text environment. In a PC, there is a lot that designers can do to make functions more visible. Following is a list of some of the more obvious ones:

1. *The PC environment uses function keys to perform some commands. Display these function keys on the bottom lines of the screen, preferably in a row to match the physical layout of the 12 function keys. If you have function keys modified with Shift, Ctrl, Alt, then the display should alter when someone holds down a modifier key. See the XTREE program for a good example of this.*
2. *What key does the user press to get the main menu? Is it F10, a slash, Alt, or some other key. Whatever key the menu is, display it on the screen at all times.*
3. *Make the user press a key between leaving the initial setup screen and entering the application proper. If you don't (and Paradox III doesn't), the sudden change of modes is very confusing.*
4. *Don't neglect the semi-graphic characters and colour when designing screens. Attractive screens might take a little longer to design, but have a major affect on the user's perception of the product.*
5. *Don't force the user to load a file, enter a filename or pick something from a menu before allowing data entry.*
6. *Follow Lotus's example, and have a line of help text accompanying every menu item. As the user scrolls down the menu, looking at the possible options, this extra line should explain more about what the selected option will actually do.*

Blank Screens

As users get more proficient, they want more screen area devoted to the work space. Ultimately, you arrive at the WordPerfect situation, when the entire screen is blank, apart from one third of the bottom line, which holds status information. This approach conflicts with the visibility required for people to initially use the software. If a designer chooses the WordPerfect minimalist approach, then

CALL FOR FREE 80 PAGE CATALOGUE

ASSEMBLERS
386 ASM Link & Locate £435.00
MASM v.5.1 £89.00
OPTASM £79.00
Turbo Assembler/Debugger £89.00

BASIC
Microsoft Professional BASIC £320.00
BASIC Interpreter £215.00
Quick Basic *U.K* £60.00
True Basic £65.00
Turbo Basic £65.00

C
Lattice C Compiler v.6.0 £160.00
Microsoft C v.5.1 £250.00
Watcom C Optimizing £245.00
C-terp £189.00
C86 Plus £310.00
Eco C88 with dev. library £55.00
Express C £60.00
High C-286 £375.00
High C-386 £560.00
Instant C £310.00
Instant C/16M £495.00
NDP C £375.00
Quick C *U.K* £65.00
Turbo C v.2.0 £89.00
Turbo C Professional £159.00
UserSoft/C £150.00
Zortech C Compiler £45.00

C++
Advantage C++ £310.00
Guidelines C++ with book £185.00
Intek C++ 80386 £310.00
Zortech C++ Compiler v.2.0 £120.00
Zortech C++ Dev. Edition £270.00

COBOL
acucobol MS-DOS £900.00
acucobol Net-Bios £1250.00
acucobol XENIX-286 £1500.00
acucobol XENIX-386 £2000.00
acucobol OTHERS CALL
Microsoft COBOL £565.00
Real Menu £210.00
RealCICS £750.00
Realia COBOL £750.00
RM/Cobol £575.00
RM/Cobol-85 £750.00

dBASE
Clipper Version 5.0 U.K. *New* £325.00
DB Fast/PLUS £155.00
DB Fast/WINDOWS £155.00
dBASE III Plus £395.00
dBASE IV £395.00
dBMAN V £120.00
dBXL Diamond £155.00
dBXL/Lan £375.00
Foxbase+ v.2.10 £245.00
Foxbase+/386 £375.00
Foxbase+/LAN £375.00
Foxbase+ Runtime £435.00
Foxbase+ SCO XENIX £495.00
Quicksilver Diamond £375.00

DATABASE MANAGEMENT
Advanced Revelation £595.00
AskSam £185.00
Clarion Personal Developer £99.00
Clarion Professional £435.00
Omnis Quartz £480.00
Opus One £240.00
Oracle for 1-2-3 £120.00
Paradox 2.0 £439.00
PFS Professional File £149.00
R:BASE Ext. Report Writer £179.00
R:BASE for OS/2 £545.00
R:BASE for DOS £439.00
Reflex £89.00

FORTRAN
Microsoft Fortran v.5.0 *U.K* £285.00
Fortran-F77L £299.00
NDP FORTRAN-386 £360.00
ProFortran £235.00
RM/Fortran £375.00
SVS Fortran £435.00

OTHER LANGUAGES
ACTOR £310.00
ACTOR Language Extension £60.00
CCSM Mumps Single User £65.00
PC Logo £90.00
RPG II Complete £995.00
Smalltalk/V 286 £125.00
Snobol 4+ £60.00
Trilogy £60.00

PASCAL
Microsoft Pascal £189.00
Oregon Pascal-2 £149.00
ProPascal £235.00
Metaware Professional Pascal £560.00
Metaware Prof. Pascal 286 £375.00
Turbo Pascal v.5.5 with objects £89.00
Turbo Pascal Professional £159.00

PROLOG
Arity Combination Pack £685.00
Expert System Dev. pack £185.00
Intelligence Compiler £299.00
MicroPROLOG Professional £425.00
MProlog £299.00
Professional Prolog II Plus £899.00
Professional Prolog £545.00
Prolog Advanced Toolkit £90.00
Prolog Compiler £120.00
Prolog Compiler/Interpreter £399.00
Prolog-86 Plus £150.00
SQL Development Pack £185.00
Turbo Prolog £75.00

AI AND OBJECT ORIENTED
C-Talk £95.00
C+ Objects Vol 1 with Source £275.00
CxPERT £245.00

COMMUNICATIONS
AdComm £185.00
C ASYNCH MANAGER £110.00
Communications Library £145.00
Comm Pak with Breakout £195.00
Multi Comm £89.00
Net Lib £125.00
NET-TOOLS £95.00
SilverComm £120.00
Comm Plus £60.00
Turbo ASYNCH PLUS £80.00

FILE MANAGEMENT
asmTREE £245.00
B-Tree with Source £110.00
B+Tree for C £49.00
AXOS £245.00
Btrieve v.5.01 Novell £155.00
Btrieve OS/2 £370.00
Btrieve Multitasking £370.00
Btrieve/N v.5.01 £370.00
Btrieve XENIX £370.00
C Index+ £245.00
c-tree £245.00
Code Base 4 £185.00
db_FILE/RETRIEVE £375.00
d-tree £310.00
dBC III £160.00
dBC III Plus £320.00
RealDL/I £599.00
WKS Library £125.00

GENERAL
ASMLIB £89.00
C Tools Plus £79.00
C Utility £120.00
C*Lib C Function Library £120.00
Clipper toolbox £239.00
DB/RA3 £125.00
dBase Program Utility £55.00
dBase Program Utility Vol 2 £55.00
dBase Tools For C £55.00
Finally £60.00
Functions with source £125.00
Integrated Dev. Libraries £90.00
Pascal Toolkit £90.00
Pascal Tools 1 £75.00
QuickPak Professional £90.00
Quick Tools Library £79.00
Spindrift Library £90.00
Super Functions £160.00
Tom Rettig Help Combo £75.00
Tom Rettig Library £60.00
Turbo C TOOLS £79.00
Turbo Plus £60.00
Turbo PowerTools Plus £79.00
Turbo Functions £69.00

GRAPHICS
dBase Tools for C Graphic £55.00
dGE v.3.0 *NEW* £125.00
EGA Toolkit £149.00
Essential Graphics £180.00
Flipper £120.00
FontWINDOW/Plus £170.00
GEOGRAPH Professional £239.00
Grafmatic CGA & EGA £135.00
GraphiC £245.00
GSS Graphics Dev. Toolkit £389.00
GSS Kernel £299.00
GSS Metafile Interpreter £179.00
HALO-88 £245.00
HOOPS £360.00
HOOPS 32 £349.00
MetaWINDOW/Plus £169.00
scLASERplus £299.00
SilverPaint £60.00
SilverPak £179.00
Slate £215.00
Universal Graphics Library £130.00

OP. SYSTEM ENHANCEMENTS
C Sharp £300.00
DESQview API C Lib/Ref £125.00
Multi C £89.00
Multi Dos Plus £60.00
OS/2 Programmer's Toolkit £215.00
OS/286 £299.00
OS/386 £299.00
Resident C £120.00
Sapiens V8 £180.00
Synergy Toolkit £239.00
Time Slicer £179.00
VMFM £115.00
Wendin OS Toolbox £60.00
Wendin PC UNIX £60.00
Wendin PC VMS £60.00
Windows Development Kit £295.00

QUERY
CBTREE/SQL £239.00
CQL Query System £239.00
dQUERY/Lib £179.00
db_RETRIEVE £239.00
XQL £480.00
Xtrieve PLUS £360.00
Netware SQL £385.00

REPORT
CB/IQ Report £139.00
Coreports £179.00
Focus Report Writer £210.00
r-tree £179.00
R&R Relational Reportwriter £95.00
R&R Clipper Module £49.00

TEXT SCREENS
Animator COBOL £120.00
Application Manager £450.00
C-Worthy v.1.1 £160.00
C-Worthy + Forms £240.00
C-Worthy + Forms + Source £395.00
C-Worthy All C compilers £480.00
C-Worthy Forms only £80.00
C-Worthy Source code only £155.00
C-Worthy OS/2 £240.00
C-Worthy + Forms for OS/2 £320.00
C-Worthy + Forms + Source £480.00
C-Worthy OS/2 & MS-DOS £315.00
C-Worthy + source £795.00
Curses £150.00
Curses Window Dev. Kit £75.00
Data & Windows £149.00
DataWindow £179.00
dWindows for db3 £60.00
Face-it £60.00
Flash-up w/Toolbox £85.00
Forms II £179.00
HI-SCREEN XL £89.00
Multi Windows £179.00
On-Line Help £90.00
PANEL PLUS £299.00
PC Forms £90.00
Pop Screen £55.00
POWER SCREEN £79.00
Pro Screen £60.00
SCREENIO £245.00
Screen Sculptor £75.00
ScreenStar £120.00
Screenplay £110.00
SoftCode £120.00
SoftCode Template £30.00
Softscreen/Help Source £150.00
Sourcewriter £605.00
UI Programmer Dev. Edition £345.00
V Screen £39.00
VCScreen £90.00
Vitamin C £135.00
VLib with Source £90.00
Vermont Views £245.00
Z View £149.00

DEBUGGERS & DISASSEMBLERS
386 Debug £125.00
Breakout-II £79.00
D'Scope 2 £159.00
dBug for dBase £125.00
DIS*DOC £79.00
Periscope II £110.00
Periscope II-X £90.00
Periscope III £875.00
Periscope I Rev 3 £435.00
Periscope IV (16 MHz) £1245.00
Periscope IV (20 MHz) £1435.00
Periscope IV (25 MHz) £1625.00
Plus Board with 512K £310.00
Plus Board with 0K £220.00
Serial Test £310.00
Soft Probe II/TX £245.00
Soft-X-plore £79.00
Sourcer + BIOS Pre-Processor £89.00

DOCUMENTORS
ASM Flow £60.00
Clear+ for dBASE £125.00
Clear+ for C £125.00
dAction £55.00
dAnalyst £139.00
Design/2.0 £215.00
dFLOW £89.00
Documenter £179.00
Flow Charting II Plus £139.00
FoxToolBox £179.00
PolyDoc £169.00

EDITORS
BRIEF v.3.0 *NEW* £149.00
BRIEF v.3.0 *UPGRADE* £60.00
BRIEF OS/2 (U.K. Version) £149.00
dBRIEF v.2.1 (U.K. Version) £79.00
BRIEF with dBRIEF £219.00
C-EDIT £239.00
EMACS £199.00
Epsilon £110.00
ME Editor with C Source £115.00
Micro ISPF £110.00
Norton Editor £49.00
PC/EDIT Plus £179.00
Personal Rexx £79.00
PI Editor £119.00
SPF/PC £149.00
Turbo Basic Editor Toolbox £59.00
Vedit+ £115.00

HELP GUIDES AND TRAINING
dBASE on-line £60.00
Norton Guides for ASM £75.00
Norton Guides for BASIC £75.00
Norton Guides for C £60.00
Norton Guides for OS/2 £89.00
Norton Guides for Pascal £75.00
Programmer's Library £239.00

LINKERS & LIBRARIANS
ALINK *NEW* £195.00
Archive Link £239.00
Link & Locate ++ £245.00
LinkLok £245.00
PLINK86 Plus *Now Polytron* £495.00
PolyLibrarian I £85.00
PolyLibrarian II £125.00
.RTLink £180.00

MAKE UTILITIES
PolyMake v.3.1 £125.00
MKS Make £99.00
MKS Make 5 User £320.00
MKS Make 10 User £515.00

PROFILERS & OPTIMISERS
Codesifter £75.00
DOSTRACE £59.00
PC Metric £120.00

PROTOTYPERS & GENERATORS
Automated Programmer £605.00
Cogen £275.00
COMPEDITOR £110.00
Dan Bricklin Demo II £125.00
Data Boss for C £245.00
Data Boss for Pascal £245.00
Genifer £239.00
Instant Replay III £89.00
Matrix Layout v.2.0 £129.00
Sycero db (U.K.) £595.00
Sycero db Net (U.K.) £745.00

SYNTAX ANALYSERS & LINTS
PC Lint v.3.0 £89.00
Pre-C £150.00

TEXT & UNIX TOOLS
C Programmers Combination £79.00
C ToolSet £59.00
MKS LEX & YACC £165.00
MKS Toolkit v.3.1 *NEW* £149.00
MKS Toolkit v.3.1 *UPGRADE* £50.00
PC Fortran Tools £99.00
PC Yacc £249.00
PolyAWK £85.00
PolyAWK OS/2 £115.00
QPARSER Plus £180.00
Text Management Utilities £75.00
Turbo Professional £75.00

TRANSLATORS
BAS_C Commercial £229.00
Bastoc £480.00
dBx dBASE to C v.4.0 £345.00
dBx Library Source £250.00
dBx UNIX/XENIX Source £470.00
FOR_C v.2.0 MS-DOS *NEW* £565.00
FOR_C v.2.0 XENIX *NEW* £630.00
FOR_C++ XENIX/386 *NEW* £1195.00

VERSION CONTROL
M.K.S. R.C.S. £125.00
M.K.S. R.C.S. 5 User £485.00
M.K.S. R.C.S. 10 User £905.00
PVCS Professional v.3.1 £395.00
PVCS Network (5 User) £1395.00
PVCS/PolyMake OS/2 £CALL

BACKUP - RECOVERY - REPAIR
dSalvage £59.00
Fastback £105.00
Fastback Plus £115.00
Mace Utilities £59.00
Norton Utilities Advanced £89.00
PC Tools Deluxe £69.00
XTree Net £235.00

CACHE
PolyBoost II £49.00

COMMS. & TERM. EMULATORS
Blast Communications £149.00
Carbon Copy Express £199.00
Carbon Copy Plus £115.00
Crosstalk Mark 4 £145.00
Crosstalk XVI £115.00
Mirror III £75.00
PC Anywhere III £85.00
Sidetalk II £70.00
SmarTerm £75.00
SmarTerm 125 £175.00
SmarTerm 220 £115.00
SmarTerm 2392 £135.00
SmarTerm 240 £199.00
SmarTerm 400 £99.00
SmarTerm 4014 £135.00
Z Stem PC-VT100 £59.00
Z Stem PC-VT220 £89.00
Z Stem PC-VT240 £175.00
Z Stem PC-VT4014 £59.00

DISK MANAGERS &OPTIMISERS
HTest/HFormat £85.00
Mace 5 £65.00
Mace Gold £95.00
Norton Utilities Advanced £95.00
SpinRite II £59.00

OP. SYSTEM ENHANCEMENTS
386MAX £49.00
386MAX Professional £80.00
DESQview £79.00
Norton Commander £55.00
Omni View £49.00
PolyShell £60.00
QDOS II £45.00
Soft Bytes 386 Max £45.00
Weiner Shell £175.00
Power Bundle £99.00
Windows/286 £59.00
Windows/386 £115.00
X-Tree Pro £75.00

FAX OR TELEPHONE CREDIT CARD ORDERS VISA OR ACCESS

PRICING
All prices exclude V.A.T., but include first class recorded delivery.

EDUCATIONAL DISCOUNTS
Generous Educational discounts available on certain products.

CATALOGUE & FACT SHEETS
Call for our new Winter '90 catalogue, over 80 pages of Software Development Tools.

.EXE 04.90

THE SOFTWARE CONSTRUCTION CO. LTD.

THE RED HOUSE, 84 HIGH ST., BUNTINGFORD, HERTS. SG9 9DJ (0763) 73455 FAX. (0763) 73460

CIRCLE NO. 842

new users will find it forbidding. Don't write your software like that - WordPerfect have been regretting it ever since. A better approach is to have function keys, menus and borders turned on from the start, but to supply options that allow proficient users to turn them off.

Menus and Text

At the heart of most programs is a menu system. These vary from the simple WordPerfect 5.0 type, to the sophisticated Windows or SideKick Plus type. They can pull down, pull up or pop up. They can be full screen, at the bottom of the screen, or at the top

When you're developing for Microsoft Windows and other such GUI systems, you have a preset list of tools which you meld together to form the menu system. This makes life easier as you don't need to code yet another menu system. It can, however, restrict you in your flexibility. If you are using one of these GUIs then you will have a style guide, such as the guides supplied by Apple for their system. Refer to these guides, to ensure that your application fits in with the general style of all applications developed for the particular GUI.

DOS allows you to have whatever menu system you like. Before you start designing from scratch, however, several de facto standards exist. These include:

- *WordStar*

The grandmother of word processors has spawned many children. It is the de facto set of shortcut keys for word processors, text editors and outliners. SideKick Plus is an example of extending the command structure to other ludicrous extremes. Recommended for shortcut keys.

- *Lotus 1-2-3*

Its layered, horizontal menus with a help line were an innovation when 1-2-3 was launched, and they still look good. The main problem is a lack of shortcuts to menus down the hierarchy.

- *COW*

The Character Oriented Window system. This is the generic name for the semi-graphic Macintosh clones such as Microsoft Works and PC Tools Deluxe. It shares the same problems of some of the GUIs: the mouse is often a necessity if you want to perform certain operations; there are a confusing number of windows; too much of the screen is taken up by menus and windows; menus at the top of the screen make it hard to keep your eye on the information in the middle.

You have probably guessed that I'm no lover of most of the GUI menu systems. So what do I prefer? Pop-up menus, preferably, available from a visible (on screen) key or a mouse button. For new users, all other functions should be visible on the screen. They should not have to use menus for simple operations such as Save and Load. Pop-ups are also easier to code, since they don't require dialog boxes and mouse support. Finally, pop-ups allow tiling of menu items to make a proper English sentence, such as Insert/Date and Time at/Cursor (where / indicates a separate menu level).

Menu Text

A typical pull down or pop-up menu item has space for three words, but it's advisable to keep it to a single word. It's very important to make that word count. In the past few years, computer jargon has led to a convention, however illogical. For example, you save your data with File Save, you paste from the clipboard with Edit Paste. It's pointless reinventing the car, so my advice is to stick to convention wherever possible. Even if the convention, like Edit Paste, doesn't make for good English. When you have a menu item not covered by convention, choose an appropriate verb.

Sometimes, to speed up menu selection, you can allow the user to press a single letter to select a menu item. This puts a constraint on your choice of menu text but it's a good constraint, because it forces you to separate text for each menu item. There are many ways you can choose your choice of letter, but following is my suggested list in order of preference:

1. *The first letter of the menu item. You should strive to achieve this as often as possible, because it makes the option much easier for people to remember. For example, Word for Windows uses File and Format for menu titles, so you need to press Alt-O to pull down the Format menu. It would have been much better if Microsoft had renamed Format to Layout.*
2. *The first letter of the second or third word.*
3. *A vowel in the first word. Avoid this method unless stuck down a coal mine, since it provides weird mnemonics that are easily forgettable.*

The Order of the Menu

The time that it takes a user to choose a menu item depends on the text, and on the arrangement of the items in the menu. Studies have shown that functional grouping of menu items decreases the scanning time for a particular item. This means you should divide the menu items into separate sections according to their function. If necessary, use a semi-graphic line between separate function groups.

You have several choices within a functional group or where there aren't enough menu items. If there is an obvious correlation, such as window number, then use that, otherwise its best to have the most frequently used items first.

Shortcuts

You place shortcuts such as Ctrl-X into your menu system to make it easier for proficient users. However, these users are very fussy, and you are bound to annoy somebody with your choice of keys.

The best solution is to allow users to redefine the keys if they want to. Whether or not you allow this, you should choose a standard set of function keys that fit in with your application. A possible scheme is shown in Figure 1.

There are many rules that you can follow when choosing the accompanying shortcut keys. Whatever you choose, though, remember to provide some logic and consistency.

Further Reading

The Psychology of Everyday Things
Donald A Norman
Basic Books
ISBN 0-465-06709-3
$19.95

Designing the User Interface
Ben Shneiderman
Addison-Wesley
ISBN 0-201-16505-8
£19.95

Apple Human Interface Guidelines
Apple Computers
Addison-Wesley
ISBN 0-201-17753-6
£12.95

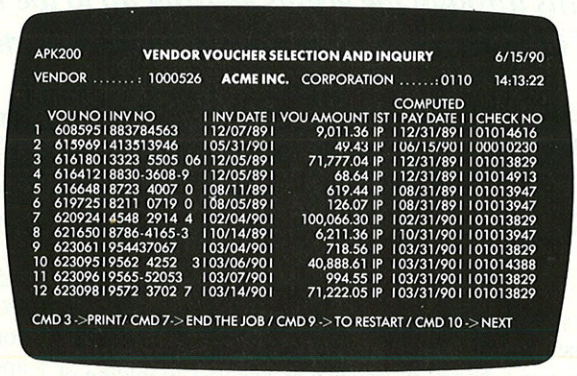
Open Look GUI Application Style Guidelines
Sun Microsystems
Addison-Wesley
ISBN 0-201-52364-7
$24.95

.EXE

Jeffrey Goldberg has eaten Raspberries while playing Cricket in a field full of Penguins for Borland. He is now stroking Fifi while feasting on Pizza at Psion.

It would take the average programmer 2 hours to find the errors on this screen.

In fact, in the 84 seconds it will take you to finish reading this ad, VERIFY could log a test script using the current version of a programme, re-run it with an updated version of the programme, and then automatically identify and resolve all errors and unexpected results. VERIFY prevents production systems surprises, immediately and interactively.



And it performs just as efficiently on a variety of CICS tests including unit, regression, stress, integration, concurrency, and migration – all of which can be performed using either an on-line or batch approach.

It makes the job easier, more accurate, faster, and much less a pain in the neck. So the average programmer could not only be finished by now, but already testing something else. So your systems would need less testing in the future.

A case of success breeding success.

The Safe Choice

Since 1969, On–Line Software has been recognised as the leading source of CICS software, education and consulting. Now we've built the same reputation in DB2. And we've been designated as an IBM* Business Partner and Application Specialist.

To protect our customers' investment in technology, our software is offered with a guarantee.

If you'd like to do less in the future, do more now– call Catrin Lloyd at our International H.Q. **01–631 3696** and ask about VERIFY.

VERIFY would be finished by now.

On·Line Software
INTERNATIONAL

Please send me more information on On–Line Software products

Name ..... Company .....

Address .....

..... Postcode .....

IBM Business Partner
Authorized Application Specialist

*IBM Is a registered Trademark of International Business Machines

Complete coupon and post to: On–Line Software International, Tenterden House, Tenterden Street, Hanover Square, London W1R 9AH
Tel: 01–631 3696 Fax: 01–493 5102 Telex: 28958 MPGLON G

.EX2

CIRCLE NO. 843

Going it Alone

Don Milne saw a gap in the PC software market, so he designed and wrote a package to fill it. Here, he runs through the events leading up to the birth of his program, from initial idea, through to the day the orders started coming in.

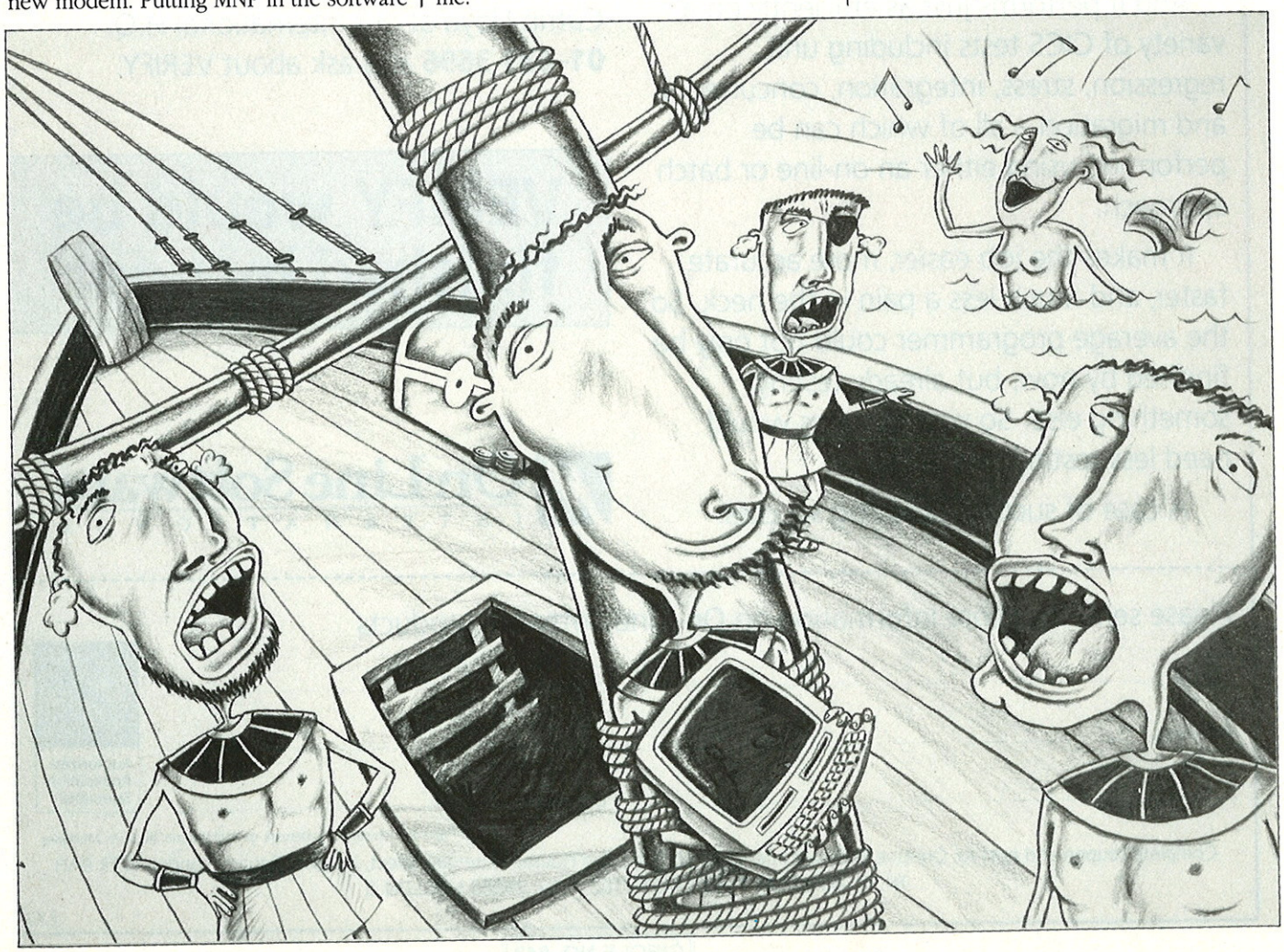
First, the small print. The program which I developed is called Odyssey, which is a communications package for the PC. The most unusual feature about Odyssey is that it provides emulation in software of the MNP error correction protocol.

Normally, this protocol is built into the modem. However, using Odyssey means that your modem suddenly gains MNP error correction, without you having to pay for a new modem. Putting MNP in the software was a brand new idea at the time my program appeared, though several others have copied it since.

If you are a programmer bored with writing specialised stock control programs for the local video shop, and feel you have the talent to try for the 'big time', but are worried about how much it will all cost, then I hope this article will give you an inkling of what it is like, or at least what it was like for me.

Backgrounds

My own background is in the Oil industry, developing specialist software for use in the North Sea and elsewhere. In late 1987, one of the companies we contract to was considering implementing a system which would have involved personnel data being communicated between offshore platforms and shore bases in Shetland and Aberdeen. Since I expected eventually to be asked to do the work, I was already mulling over the



technical difficulties in my mind. First of all, the most obvious way to transfer the data would be by modem, which introduced the likelihood that line noise corruption would occur. It would be nice if the companies could be persuaded to buy one of the new modems which supported an ARQ protocol (Automatic Repeat on reQuest - a generic name for error correction protocols such as MNP and V42). However, the odds were that they would do what most big companies seem to do, which is to make a beeline for the nearest supplier of market leading, overpriced and underfeatured products. In other words I could expect to be working with a 1200 bps modem, with no error correction, and if I was very lucky, it might accept a subset of the Hayes command set (the de facto standard for intelligent modem control). Not something to look forward to.

It was then, with a great deal of pleasure, that I discovered an archived file on a bulletin board (Deep Thought, now sadly no longer with us), which claimed to implement the MNP protocol in software on a PC. If I could get this working then it wouldn't really matter which modem the company went for, because I could add the error correction myself, in a way which would be compatible should they go for better modems later.

Once I had the file on my PC, my enthusiasm began to dim. The software was in a sad state. It took the form of a C library, with a smattering of assembler. Some modules were missing, others would not compile, nowhere was there a description of the protocol itself, and the general quality of programming was frankly rather poor.

Just as I was about to give up, someone on CIX (another UK BBS) came to my aid. He worked for IBM, and had just been given a copy of a submission to the CCITT by Microcom (the creators of the MNP protocol) in which there was a complete, clear description of all the MNP layers up to class four. Armed with a copy of this, I was able to write my own MNP engine, from scratch, and this time it worked.

What to do next

At this point I had no thoughts of turning my minor success into a commercial product. All I had been doing was satisfying my interest in a technique, merely adding another weapon to my armoury of software tools which could be called on should I need them. However, several people on CIX had helped me gather the information I needed, and so to pay them back I uploaded a little program - a simple, no-frills, terminal emulator which made use of my MNP emulation.

I was amazed by the response. Many people, including several well known journalists (this was important later), said that they were impressed with seeing MNP in software, and why didn't I take it further - why not build a complete comms package around my little program? Now that I was finally confronted by it, I could see that they were correct. If I could do a good job of creating a complete product then I ought to have a winner on my hands. MNP in software was a genuinely new idea, and genuinely new ideas are too few in the software business to be passed up lightly.

Persuading the Boss

The next problem was setting aside enough time to work on a complete comms package. The firm I worked for was very small (half a dozen of us), so losing the resources of one of their senior programmers for several months was quite a load to bear. Also, the type of work was quite outside any of our commercial experience. All the other work the company did had a purchase order up front (and, therefore, a promise of payment).

Do not kid yourself about how much development of a product like this costs. Even doing it on the cheap, you can expect any significant software product to require a five figure sum in development costs, and that's before you get into distribution and marketing. As a simple example, your employer loses whatever he pays you in salary, plus whatever he normally charges as a day rate for your time while you are unavailable for normal work. Add up the figures for yourself - and you have only accounted for personnel costs so far, for one body.

Despite this, I came to an agreement with the managing director that any pressing mainstream work would be dealt with as needed, but at other times I could work on my project. As at so many stages of Odyssey's development (it was around this time that I chose the name), my participation on CIX was vital. I could show my boss the feedback from CIX concerning my demo program, and that collection of journalists was what persuaded him. At least we could be sure that when it came time to present the product for review ('We're Micropack, here's a great comms package'), the answer was not going to be 'Who?'.

Choosing the Tools

During the first month of development I chose my tools, and worked on the fundamentals - getting reliable serial I/O, and getting the bugs out of my MNP engine. I chose to continue work using my favourite language, Modula-2, and was greatly aided by the fact that my interest in MNP had coincided with the appearance of an excellent compiler for that language, JPI Modula-2, which I had helped to beta test. Like C, Modula-2 is designed for low level work. It's modular (again, like C) but is even more strict in type checking than Pascal, features which in my opinion make this language an ideal choice for development of a comms package, where efficiency and robustness are so difficult to achieve, but so vital.

I did make one about-turn during the early design stages; my initial idea had been to produce very much a Procomm compatible, with the MNP as an added extra. Procomm is a popular shareware package, still one of my favourite PC comms programs. However, it soon became obvious that being Procomm compatible meant carrying a lot of baggage that I didn't need. For example I particularly wanted to keep the Procomm script language, so that all those ex-Procomm users like me would not have to learn a new one. However, keeping complete compatibility with the script language meant support for all the features of Procomm accessible from the script language, many of which I disliked or were inapplicable.

Eventually I had to bite the bullet and decide that compatibility with another package was too restrictive. I designed my own interface, which retained some of the flavour of Procomm, without being identical, and scrapped the Procomm script language entirely. For a replacement language I decided to go with the style I liked - a style very much like Pascal or Modula-2. In fact the syntax I chose is almost exactly that of M2, with some concessions where I felt that M2 was unsuitable for beginners (for example the script language is not case sensitive, and allows implicit type conversion, and strings are directly supported in the language). In fact I borrowed shamelessly from several languages: overall structure from Pascal and C; statement syntax from M2; certain C features such as the ability to ignore the return value of functions when you don't need them.

Throughout this time I made use of the comms conference on CIX, putting up test programs and dealing with feedback. One of the main problems I faced was the lack of truly in-depth technical information about the PC - the sort of stuff you don't find in any IBM reference manual or programmer's guide. CIX met my needs, by providing a large number of users with a

wide range of differing hardware and software configurations. Obscure problems presented themselves, their cause was investigated and diagnosed, and new revisions of the test software were made available. I believe that my access to CIX made the confirmation and correction of these sorts of problems far faster than could have occurred any other way, even if I had had the resources of some mega-outfit to work with.

Adding Features

If there was any dominant trait in the design process it was this: I loath featurism. I am determined not to add frills simply to copy the competition. Quite often a question I would get from a new tester would go 'Why hasn't it got X? - program Y has it'. I would then reply, 'Ok, tell me why you need it?' and, more often than not, it would turn out that they themselves never used that feature, but it might be a good idea anyway. In my opinion, features like that are like go faster stripes on a car - it might make the product look impressive, but does that make it more useful? Examples of this were: emulations of terminals no one ever uses; support for a million file transfer protocols that no BBS supports, and so on.

I have to admit that my opinions were not always correct. For example, I was initially against having a dialling directory. I felt that this was an area of confusion in Procomm that I could well do without, and that automatic dialling was easily dealt with in the script language, and if you didn't want to mess with the script language I had provided a 'learn' mode to create scripts for you. It was eventually brought home to me that the learn facility is not perfect, which meant that naive users sometimes were forced to mess with the script language, just so that they could get a comms package to dial a number. Eventually I had to eat humble pie and implement that feature (now I would never go back - handy things are dialling directories).

Committee Stage

It may have crossed your mind that what I was doing was that most scorned by methodologies, design by committee. This was never true. The design at every stage was my own, although I did (and still do) accept input from participants in the process if I considered the idea a good one. In every case however, I would develop a design idea either on my own, or in response to a user's difficulty in some area or other, I would implement that idea, and make a new version available on CIX for testing and later refinement.

As I got closer to having a complete product on my hands, I closed down the open area on CIX that I had been using, and set up a special closed conference. The CIX system does not publicise the existence of a closed conference - the only people to know about it are those who are invited, by the person who set it up - to join. I invited beta testers and, as expected, got replies from many of the people who had helped in the early experiments. At this point I had the luxury of accepting people who had contributed often to the initial testing stages, while rejecting the technically naive and the freebie-seekers who had contributed nothing.

Many people said that they were impressed, and why didn't I build a complete package around my little program?

The wastage of standard forms of beta testing of the type that big companies like Borland or Microsoft use appals me. Whatever product you need to test, it makes a great deal of sense to use some form of electronic mail. If this means that you have to limit your beta testers to those that possess modems, this, in my opinion, is a small price to pay. The speed and ease with which you can get feedback, and distribute new versions of the code, save huge amounts of time and money.

Doing the Manual

In parallel with the later stages of development on CIX was the development of the user manual. We wanted a manual that reviewers would be complimentary about, so it had to be well written, and have good contents and index pages. In other words we wanted exactly the sort of manual that is not written by the programmer. This led us to our biggest mistake: we hired a 'technical author' to write the manual, because I was tied up with the program itself. We were persuaded of the person's abilities by examples of his work printed in various computing magazines. They were not very prestigious magazines to be sure, but then, as a small company, we could not have afforded the services of a prestigious author.

I remember reading the first draft in a plane while I was off on a trip. The manual was awful. The English was terrible, and worst of all, much of the information given was just plain wrong. A first draft it may have been, but this looked like it really needed to be rewritten from scratch. I suppose that it was our fault, for picking the wrong person for the job, or for not keeping tighter control, but why on earth had he not asked for confirmation of some of the technical information, instead of just making up some rubbish? At this point, we were a month away from the deadline - ads had been placed in magazines, and we had better be ready to ship when the orders started coming in.

Patches

In the end, all we could do was patch the thing up. I went over the whole document, correcting all the errors of fact, and tidied up as much of the English as I could, but even so it was a rush job, because we still had to get it typeset and printed. We did not even have time to generate an index (you need the page numbers for that, which means waiting for the typeset copy, and by the time we had that we had to ship NOW). I make no secret of the fact that I consider the manual the weak link in Odyssey. The only happy thing about it is that users tend not to read manuals anyway. Those that do, seem to take it as par for the course that software manuals are pretty bad. One day I will have to rewrite that manual completely, but that will have to wait.

Eventually the deadline arrived, and shipping we were, ready or not. In total, the development of Odyssey had started in earnest in May 1988, and we shipped in October 1988. It took a further year to recover our development costs. During that year we took the step of licensing the full MNP protocol from Microcom, allowing us to implement the more efficient class five, but delaying the break even point. Class five MNP includes built-in data compression, which can increase the throughput of a modem by almost 100% on pure ASCII text files.

Marketing

Marketing is a difficult problem for a small firm. Advertising is hugely expensive, and we did not expect the returns to justify a big advertising splash, even if we could have afforded it. For one thing, we were of the

opinion that the only worthwhile ad was a full page colour spread in a popular magazine - a teeny little box in the corner of page 99 of 'PC NeverHeardOfIt' was going to be noticed by nobody. However, full page colour ads in a worthwhile magazine cost at least £1000 per page, per issue (maybe the big boys can negotiate - we couldn't).

In the end, we decided to dip our toes in the advertising waters. Three full-page ads in consecutive issues of 'Computer Shopper' were placed. The response to these ads was poor. In fact, we did not even cover our costs (advertising, not development). I'm sure that things would have been better had we consulted experts, but I doubt that we could have afforded their advice. Frankly, we have come to the opinion that there is no cheap way to advertise - if you are a small software house, you have to find your market another way.

The advertising did bring in a trickle of customers, then the press mentions started to appear, followed by the odd review. I have to say this carefully (I don't want those concerned to think that I cultivated their acquaintance simply for what I could get out of them, or think I am accusing them of favouritism), but I feel that my CIX contact with several important computer journalists was invaluable in not only getting reviewed at all, but favourably reviewed. I think it made a difference that the product they were reviewing was not from some faceless outfit they never heard of, but was the work of a person with whom they had daily contact. Also, the product was one that they themselves had had an important influence on, prior to release. The opportunity for journalists in the computer press to have contact with professionals in the computing game over the BBS networks is one of the most positive advances in years.

The reviews were not always glorious. Although we never had a bad review, it was noticeable that with those reviewers which did not know me, some of the remarks seemed to be more grudging respect than real enthusiasm - there seemed to be a strange reverse chauvinism going on which said that any US product, however poor, had to be better than a product, however good, from a small UK company.

You could also tell when the reviewer was an enthusiast, rather than a journalist committed to objectivity, when in the summary their personal favourite would receive star rating ('this has everything'), while our product would only be 'good', despite having all the features of the other product, implemented in an almost identical way, PLUS the extras such as MNP, a better script language, the Zmodem file transfer protocol and so forth. I still find that tendency mystifying, but I have to admit that we have been very lucky that the reviews so far have been kind to us. Our poor manual could potentially have meant disaster.

As each new review appeared, our trickle of customers increased, until eventually we hit the jackpot - a review in PCW. Then the trickle turned into a rush. That meant a rush of customers too, which brings me conveniently to the subject of support.

Support

Support is a problem area for a small company. How could we handle all those customers we had attracted? Since Odyssey

was my creation alone, I am the only one around who can answer the really obscure questions that come up. On a personal front, I must be honest and admit that having my working day interrupted perhaps a dozen times can be irritating, but I suppose I should look on it as the price of success - I just have to take a deep breath, and put my voice into jovial mode.

Obviously, the best way to handle support is to design the package as much as possible so that support is not necessary. This means that you have to bear support problems in mind before implementing any new feature. On several occasions, I turned down suggestions for changes to the program on the grounds of the potential for confusion it would cause, and because it would be me who would have to suffer the consequences.

However, the nature of the product makes this ideal easier said than done - we picked the wrong subject (comms) if we wanted to minimise support requirements. On the other hand, the majority of comms users seem to be more technically knowledgable than your average business person, which probably compensates a lot.

In general I try to be friendly to the people who call me, and they in turn are friendly to me. In all the time that I have been supporting Odyssey, I have received exactly two letters from annoyed customers, in both cases because the customer concerned had assumed that certain features would be supported which were not - but had not checked first.

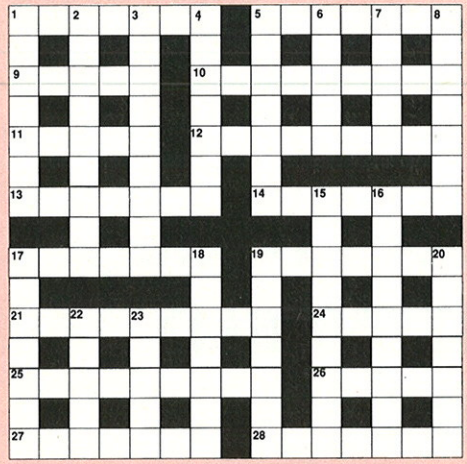
The first of these customers had one of the early versions without a dialling directory, and rightly criticised the package because of it. I supplied him with a new version, advised him that being friendly would cost him nothing, and might gain him something (like the new version which I enclosed), and everything was all right after that. In fact I still have occasional (friendly) electronic mail correspondence with the same person.

The other letter was a different matter. Again it was over some feature that he assumed would be there (support for split speed V.23 modems). The letter came close to accusing us of selling under false pretences, and demanded that we right the wrong or give him his money back. It was quite apparent that he felt his giving us money entitled him to use this kind of abuse. Frankly, I can do without people like that, so in my reply I told him that refunding his money was a great idea. If he had been friendly about it, perhaps I might have done something to help, but if someone is going to be abusive, we'd rather survive with one less sale.

.EXE

Don Milne can be contacted, care of Micropack, on 0224 631100, from whom Odyssey currently costs £70. We at .EXE would be very interested to hear from other companies or programmers who wish to write a similar 'how I did it' article . The full specification of MNP (not class five, though) is in the public domain, and should be available from a bulletin board near you. The CIX office should be able to send you a disk for around £3, if you don't have access to a modem. The CIX office is on 01 390 8446.

.EXEWORD



ACROSS

1 Software tool supplied in wartime? (7)
5 Little Sidney comes back, admits, gives up (7)
9 Sets of punched cards he mainly floors (5)
10 Air due to breaking up displays (9)
11 Belonging to them? Not here, I hear (5)
12 Spoil a signal with noise maybe (9)
13 Sorting with a mesh (7)
14 Biting mis-reported with a +/- symbol (4,3)
17 On average half a disc spin cycle (7)
19 When the last errors may appear (3,4)
21 Try to sign up Rog for some collections (9)
24 What pointers do to and fro (5)
25 Tele-commuters harking back to the past (9)
26 Provide education with a series of bits (5)
27 Trots along to take 3.14... back out of sweets (7)
28 They may use spells or bubbles maybe (7)

DOWN

1 Ape stud who comes round as he merges (7)
2 Ten NE micr devices with a little bit extra (9)
3 Add charged particle into set or list maybe (9)
4 Such a gap is a sign of boredom (7)
5 Symbolises something of French paper money (7)
6 Thoroughly clean the largest university in a register (5)
7 To store data is not wrong it seems (5)
8 Partition a billion people in a group (7)
15 Currently produces, reports ... (9)
16 ... and where they're stored for carriage (a small example) (9)
17 I follow old calculating number with educational computing, basic to IT (7)
18 Americans pull the diocese back (7)
19 Tries to block the signal over masks (7)
20 19 ac jobs may go wrong with simple gates (7)
22 Number base puts short month on metal (5)
23 Flat beginning for article in mid-East groups (5)

Apologies for printing the wrong solution last month. To make amends, we are printing the solutions to February's and March's this month.

FEBRUARY

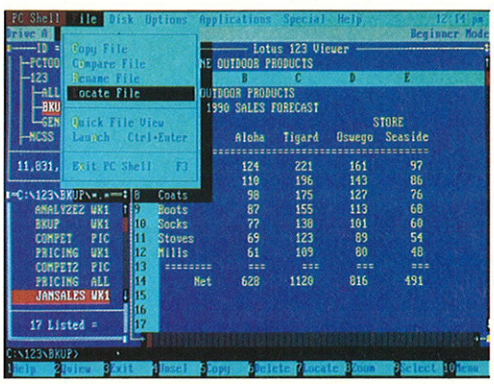
P	A	R	I	T	Y		A	C	C	E	P	T	O	R
A		E		A				A		N		W		E
R	I	G	O	U	R		P	L	O	T	T	E	R	S
A		I		T				I		R		E		I
L	I	S	T	E	D		A	F	F	E	C	T	E	D
L		T		N		S		O		E		S		E
E	L	E	M	E	N	T	A	R	Y					
L		R		D		E		N		B		S		S
					O	P	T	I	M	I	S	T	I	C
A		E		F		B		A		S		O		U
D	I	S	P	L	A	Y	S			A	T	O	N	A	L
D		S		A		S				A		I		P
E	V	A	L	U	A	T	E		O	B	J	E	C	T
N		Y		N		E				L		S		E
D	E	S	K	T	O	P	S		N	E	S	T	E	D

MARCH

C	O	M	P	U	T	E	R	I	S	A	T	I	O	N
O		A		S		M		C		B		C		E
N	O	I	S	E		O	P	E	R	A	T	I	N	G
S		N		R		T		B		N		N		A
O	B	S	E	S	S	I	V	E		D	I	G	I	T
L		T				O		R		O				E
E	R	O	S	I	O	N		G	E	N	E	S	I	S
		R		M						E		P		
T	H	E	R	M	A	L		F	I	D	G	E	T	S
O				E		E		O				C		E
F	I	E	L	D		N	O	R	M	A	L	I	S	E
F		F		I		G		T		L		F		K
E	X	T	R	A	C	T	O	R		G	U	I	D	E
E		A		T		H		A		O		E		R
S	Y	S	T	E	M	S	A	N	A	L	Y	S	I	S

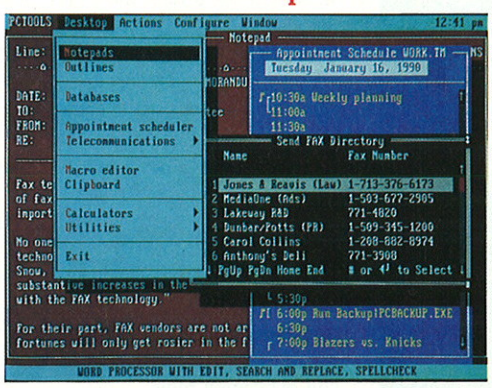
About. PC Tools Version 6.

DOS Shell



Three user levels let you start with the basics and add more advanced features when you're ready. New features include over 30 file viewers, configurable function keys and resident DOS prompt. Network compatible.

Desktop



Features fax support, dBase-compatible database, notepad, calculators and appointment scheduler. You can automatically schedule such tasks as regular backups and performance enhancement programs.

improve your productivity and protect your data, simply return the coupon or call us.

And get your hands on the utility that everyone is talking about.

CIRCLE NO. 844

PC Tools Deluxe 6

Please send me: Version 6 details ☐ *Upgrade details* ☐

Name _____ *Job Title* _____

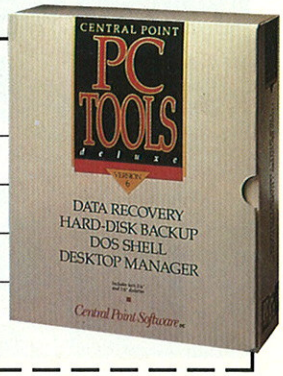
Address _____

_____ *Post Code* _____

Tel: _____ *Current Utility used* _____

Central Point Software
P.O. Box 15 Twyford Berks RG10 9LZ *EUROPE LTD.*

.EXE 4/90



WHERE THE SOFTWARE DEVELOPMENT WORLD COMES TOGETHER

Exhibition and Conference

12-14 June 1990

Wembley Exhibition and Conference Centre

Software Tools is the UK's only exhibition to showcase the *whole* variety of tools and technologies available to today's software developers. And rapid market expansion means that this year's show will be bigger than ever before. Over 140 key suppliers and manufacturers will be launching and demonstrating the very latest tools and techniques available, from analyst workbenches, CASE tools and expert systems through to re-engineering, prototyping systems and debugging tools.

To take advantage of this complete software tools package, pre-register for Software Tools '90 now. Simply call Lynne Davey at Blenheim Online on 01-868 4466 or clip your business card to this ad and return it to her at the address below **before 11 May 1990** to be sure of saving the £5 entrance fee.

BLENHEIM ONLINE
The IT event specialists

Blenheim Online Limited
Blenheim House, Ash Hill Drive
Pinner, Middlesex HA5 2AE
United Kingdom

Phone: 01-868 4466
Fax: 01-868 9933 or 01-866 5691
Telex: 923498 ONLINE G

E05.2

CIRCLE NO. 841

The First One

In 1954, there were fewer than 10 computers in Britain, all programmed in pure machine code or a simple assembly language. The Manchester Autocode broke new ground by providing a user-friendly programming language for the first time. Martin Campbell-Kelly explains.

To appreciate the impact of the Manchester Autocode in the 1950s, it is necessary to understand the computing environment for which it was created. The modern digital computer was actually invented in America in 1945, and was first described in the classic *First Draft of a Report on the EDVAC* written by the great American mathematician, John von Neumann. Soon after the EDVAC Report was published, computer laboratories in the United States and Britain began to work on building the new type of machine.

In 1948, Professor Freddie Williams and Dr Tom Kilburn of Manchester University secured a world lead for Britain by being the first research group anywhere to build a working computer. Although their 'Baby Machine' was a tiny computer - just 32 words of memory and no peripherals - it established incontrovertibly the soundness of what has now become known as the von Neumann Architecture. During 1948-49, the Baby Machine was developed into a full-scale prototype computer with a thousand words of random-access memory (based on CRT storage tubes) and a magnetic drum of several thousand words.

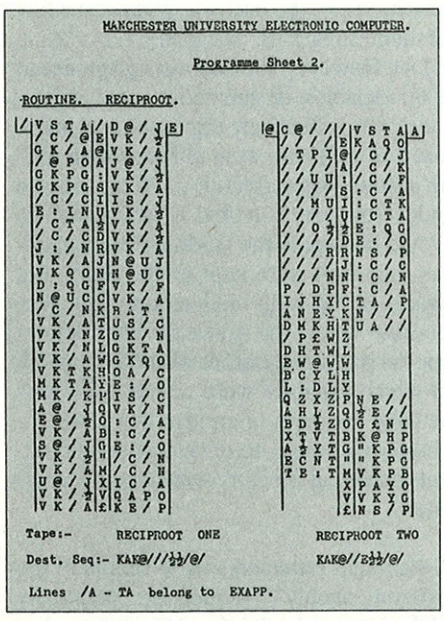


Figure 1- A square-root subroutine for the Manchester Mark I computer. Note the absence of comments!

Williams and Kilburn were essentially electronics engineers, however, who were more interested in building machines than in using them. The University, therefore, persuaded the famous British mathematician, Alan Turing, to become Assistant Director of the laboratory and to take the lead in developing the use of the computer for applications and mathematical research.

One of Turing's first tasks was to develop a programme system for the Manchester machine. Unfortunately, he designed a system that was both primitive and exceedingly difficult to use. While many computer centres were to develop programming systems based on hexadecimal or octal, Turing decided to go one better and use base-32 arithmetic. (This was because each 20-bit word of the Manchester computer was made up of four 5-bit characters.) Turing's decision meant that users of the Manchester computer had to get used to what was, in effect, base-32 arithmetic, written backwards, and in which there was no correspondence between each digit and how it was written.

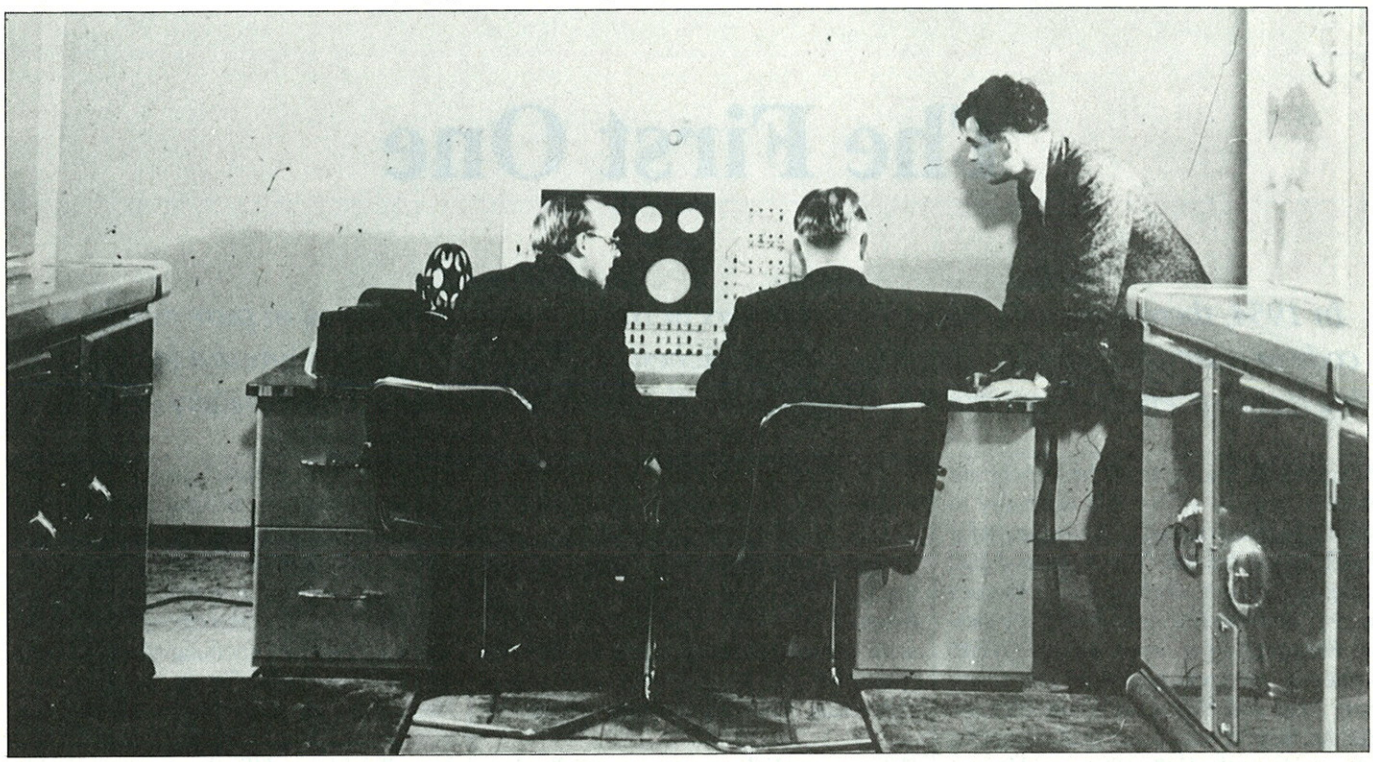
Figure 1 shows an example of a library subroutine for the Mark I written by Turing. In addition to the awkward base-32 arithmetic, there was no relationship between the written form of an instruction and its function (such as, for example, using 'ADD' or even 'A' to represent an Add instruction); furthermore, there were no symbolic names or labels; there were no data-defining operations; subroutines could not be relocated and so on. Add to this the fact that programmers were not then always in the habit of writing comments against their code, and it is easy to appreciate why programming for the Manchester computer was regarded as something of a black art. There is no doubt that in designing the Manchester programming system, Turing was handicapped by his great intellect: he himself had no difficulty in coping with base-32 arithmetic written backwards, and he saw no reason why anyone else should.

The Ferranti Mark I

During 1949-50, a properly engineered version of the Manchester computer was built by the local electrical engineering firm of Ferranti, which was duly installed in a newly-built computer laboratory in February 1951 (Figure 2). The Ferranti Mark I was the world's first commercially-manufactured computer to be delivered; it cost in the region of £65,000 (the equivalent of about £1 million today), contained nearly 4000 vacuum tubes and was capable of performing about 1000 operations per second. The arrival of the new machine established in the University a world-class computer laboratory; this was a facility which was to prove a major technical resource, not only for the University, but also for many big engineering firms in the North West of England, as well as for aerospace firms and the oil industry.

By this date, Alan Turing had lost any interest he ever had in developing programming systems, having become deeply involved in using the computer for research on morphogenesis (the growth of living forms). By default, this role as the leader in programming was taken up by a young mathematician, Tony Brooker (who later became the University of Essex Professor of Computer Science).

Figure 2 - The Ferranti Mark I computer delivered to Manchester University in February 1951. The figure to the right of the console is Alan Turing



So that users from other departments of the University and local industry could make use of the computing laboratory, it became necessary to organise short courses in which they could learn to program the machine. The main brunt of organising these courses fell on Brooker, and he was dismayed to find that it took about two weeks for a novice to learn programming, and very much longer to become reasonably proficient at it.

A Simple Machine

It was not just the awkward notation that troubled novice programmers, though that was bad enough. A second difficulty was that the machine lacked floating-point hardware, which meant that programmers either had to organise their calculations very carefully or accept the overhead of using floating-point subroutines. A third difficulty was that of organising the flow of data between the random-access memory and the magnetic drum: in a big program just organising the data flow could easily dominate the complexity of the original problem itself.

Because it took so long to learn to program the machine in the first place, and then to code and debug an application program, there was a real disincentive to write anything but serious 'production' programs - that is, programs that could be written once but used many times. Thus, the laboratory had a hard core of professional users writing big matrix and linear programming suites; but the needs of 'amateur' programmers who wanted to undertake exploratory or one-off calculations were not being catered for at all.

It was for this class of user that Brooker devised a very simple programming system, that he initially called 'The Simple Machine', but later called 'Autocode'. The Autocode was a programming system that would enable users to write programs quickly, by hiding all the details of the basic machine code, floating-point arithmetic and storage organisation. Thus the system fulfilled exactly the same objectives that the BASIC language was to meet a decade later, and for essentially the same reasons.

The Autocode

The Autocode was a very simple programming language; not least for the reason that Brooker wanted to be able to describe the entire system on two sides of paper, with a third to describe an example. Figure 3 shows .EXE's famous Triangle Problem, coded in the style of the Manchester Autocode. Although it is easy to criticise the syntax and semantics of the system with the benefit of hindsight, it must remembered that even the use of terms like 'syntax and semantics', not to mention 'programming language', then lay well in the future. The Autocode was an artefact of its period - designed more or less from scratch and with few significant precedents.

Like most programming languages, the Autocode recognised two kinds of number: integers and reals; but, since that particular terminology was also not to emerge for several years, they were known as 'indexes' and 'variables'. An index, of course, also corresponded to an index register, although the user was entirely shielded from the underlying mechanism. So that all the indexes could be kept in the random-access memory, the user was given just a few of them (in fact 18: designated `n1`, `n2`, ..., `n18`). However, the user was given about 5000 variables, designated `v1`, `v2`, `v3` ...; these were effectively organised as a giant-sized array. These were either written with an absolute subscript (eg `v23`) or using an index (eg `vn7`). A few of the variables would be kept in the random-access memory, but most were kept on the drum. But again, all this was hidden from the user, because within the Autocode interpreter a special paging algorithm would keep track of which variables were where. (This was actually the germ of an idea that led Manchester University to develop the first patents on virtual-memory systems a few years later.)

Assignment statements were simple in the extreme, mainly to reduce the complexity of the interpreter. It is hard for us to realise today that compiling an arithmetic expression with brackets and subscripts was a significant research problem in the early 1950s. At most, an expression could contain a single operator and two operands, or a

FTN77/386

FULLY FEATURED FORTRAN COMPILER FOR YOUR 386

FEATURES, FUNCTIONALITY AND OUTSTANDING PERFORMANCE COMBINE TO MAKE FTN77/386 THE FIRST CHOICE FOR PROGRAM PORTING AND DEVELOPMENT

- ☐ **The FASTEST PC Fortran compiler around** – 13,000 lines per minute on a 16MHz IBM PS/ 2 Model 80.
- ☐ Compiled programs exploit the 386 hardware FULLY, enabling you to address up to 4 Gigabytes of memory.
- ☐ Full VIRTUAL MEMORY capabilities.
- ☐ ANSI Fortran 77 plus mainframe extensions.
- ☐ Support for 80287, 80387 and/or Weitek coprocessors.
- ☐ Coprocessor emulation for non-numerical programming.
- ☐ Runs under MS-DOS.
- ☐ Optional global optimisation *or* checkout mode with diagnostics.
- ☐ Full window-based source level debugger.
- ☐ Currently the *only* 32-bit 386 PC compiler to appear in the NCC list of ANSI validated Fortran compilers.
- ☐ Supplied with a comprehensive, fully indexed reference manual.
- ☐ Allows libraries to be linked dynamically to programs during execution. Dynamic linking means that EXE files are kept small.
- ☐ Contains a full 32-bit in-line symbolic assembler with window-based debugger.
- ☐ Interface to real mode programs and libraries.
- ☐ Selection of graphics libraries available.
- ☐ NAG Fortran Workstation Library and NAG Graphics available.
- ☐ Fully integrated with the DESQview/386 multi-tasking/ windowing software.
- ☐ Novell Btrieve interface.
- ☐ Provides a library of over a hundred routines for MS-DOS access, non-standard file manipulation, window management, graphics, etc.

SALFORD SOFTWARE MARKETING LTD
COMPUTING CENTRE · UNIVERSITY OF SALFORD
SALFORD · M5 4WT · ENGLAND
Telephone (+44) 061 745 5678
Telefax (+44) 061 745 5666

FTN77/486 AVAILABLE NOW!

single function call. For example, the statement `v2 = 1.0/v3` would set `v2` to be the reciprocal of 3; and the statements `v2 = F5 (v2)` would replace `v2` by its square root (the function `F5` corresponding to the square-root subroutine).

Input/output was minimalist. To print the result of a computation, an asterisk was simply placed in front of the corresponding assignment statement. This was very useful for debugging, because asterisks could be put *ad lib* in the program to produce diagnostic information, and then edited out once the program had been debugged. To read in a numerical value, the function `I` was used; thus `v2 = I` would read in a number and assign it to `v2`.

Of course, the price to be paid for this simple scheme was that input/output was primitive, even by the standards of the day. For example, it was not possible to print table headings, or to read or write alphabetic information. (In the Triangle Problem, the different cases have been indicated by the code numbers 1, 2, 3 and 4 instead of informative captions.)

In 1954, the idea of goto-less programming was many years in the future, and, in fact, the only Autocode control mechanisms depended on using jumps. Statements were labelled with integers and an unconditional branch to statement number 3 (say) would be written `j3`. A conditional branch statement such as *jump to 3 if v3 exceeds 24* would be written `j3, v2 > 24`. Finally, to terminate the program, the halt instruction, written as `H`, was used.

Equipped with this information it should now be possible to understand the Triangle Problem in Figure 3 in its entirety. And, in fact, there is virtually nothing more to learn about the Autocode. A Simple Machine it certainly was.

Aftermath

Brooker developed a prototype Autocode system in early 1954 and started to use it during that year on an experimental basis, with astonishingly good results. A new user could be trained in the Autocode on a half-day course, instead of the two weeks it previously took to train a machine code programmer. Instead of it taking several days for a user to develop and debug a program, it would take not more than an hour or so. Of course, there was a hefty penalty in the overhead of the interpreter; but for small, one-off programs, this was quite acceptable.

Machine time using the Autocode gradually built up from about an hour a day at the beginning of 1954, to three or four hours a day by the end of the year. The following year, a postal service was started so that remote users - such as chemical engineers at ICI in Middlesborough and researchers in the Atomic Energy Research Establishment at Harwell - could use the system. The postal service alone soon accounted for 12 hours of machine time a week. One day in 1956, Brooker was buttonholed by an excited user from a large engineering concern who told him that the Autocode had just made his firm several thousand pounds because they were able to do some crucial engineering calculations at a speed that was never previously possible. It was at that moment that Brooker realised that programming languages were to be the way of the future.

By 1957 or 1958, the Autocode principle was so firmly established that virtually every British computer manufacturer was offering a programming system more or less closely based on the Manchester Autocode. For example, Ferranti offered the Pegasus and Mercury Autocodes; Elliot offered the 803 Autocode; English Electric had the DEUCE Alphacode; and there were plenty more.

However, by the early 1960s, FORTRAN had established itself, in the United States, as the standard programming language for second-generation computers and, in Europe, Algol 60 was seen as the way forward. The economic advantages of a standardised programming language in terms of application program portability were overwhelming; and by the mid-1960s, the era of the British Autocode was past.

.EXE

```
1    n1 = 1                    ┐
2 *  vn1 = I                   │  Read and print the three sides
     n1 = n1 + 1               │  of the triangle, v1, v2 and v3
     j2, 3 ≥ n1                ┘
     v4 = v2 + v3              ┐
     v5 = v1 + v3              │  Test No. 1: Triangle Inequality
     v6 = v1 + v2              │
     j3, v1 > v4               │  Print '1' and halt if the lengths of the
     j3, v2 > v5               │  three sides do not represent a triangle
     j3, v3 > v6               │
     j4                        │
3 *  n2 = 1                    │
     H                         ┘
4    j5, v1 ≠ v2               ┐
     j5, v1 ≠ v3               │  Test No. 2: Equilateral Triangle
   * n2 = 2                    │
     H                         │  Print '2' and halt if the triangle
                               │  is equilateral
5    j6, v1 = v2               ┐
     j6, v2 = v3               │  Test No. 3: Scalene Triangle
     j6, v3 = v1               │
   * n2 = 3                    │  Print '3' and halt if the triangle
     H                         │  is scalene
6 *  n2 = 4                    ┐
     H                         │  The triangle is isosceles: print '4' and halt
     (j1)                      ┘  Start program at label 1
```

Figure 3 - The Triangle Problem in Manchester Autocode

Dr Martin Campbell-Kelly, himself a graduate from Manchester University, lectures in Computer Science at the University of Warwick, specialising in the history of computing. He has published many works on this subject; most recently he has edited the Collected works of Charles Babbage, and has written ICL: A Business and Technical History (Oxford University Press, 1989) - the official history of Britain's leading information systems firm.

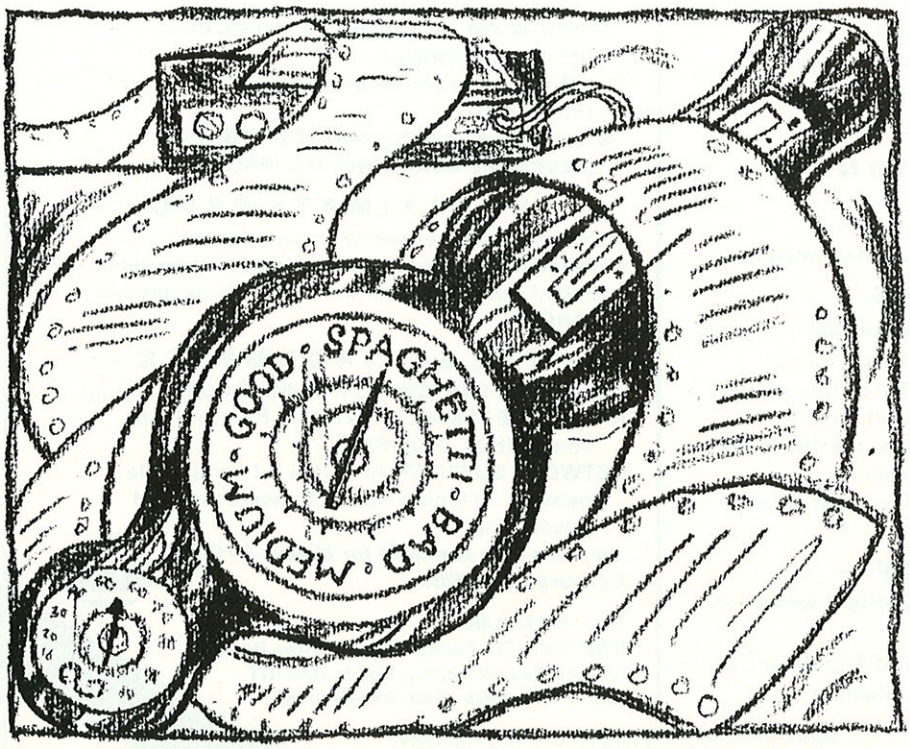
How Good is Your Software?

It is easy to judge the design of software if you have to re-code it twice a week. But Darrel Ince can tell before the program is written, as he explains.

In the early 1980s, two American academics based at the University of Iowa, Sally Henry and Dennis Kafura, spotted something very strange about UNIX. They were researching software metrics - numerical measures of the goodness of a computer system - and had formulated a metric, based on the program code, which they hoped would indicate the quality of the modules that made up the UNIX operating system.

They examined all the modules, and calculated the metric for each one. They plotted their results as a graph: the modules' metrics against the amount of resource expended on each module in correcting errors. The graph they produced was a straight line, showing that, as their metric increased, the amount of effort tracking down errors and modifying UNIX increased linearly. However, one module did not satisfy this relationship. Most of the modules had a complexity metric value ranging from 10 to 100. This module had a complexity of 108 - although the software maintenance team had done very little work on this module. The explanation soon became clear. The module was so complex that no programmer dared touch the code. It was the spaghetti junction of the UNIX system, a programmer's no-go area.

The Kafura and Henry work was of major significance. They showed that it was possible to attach numbers to a system that gave you an idea of the quality of the design. They were also able to identify potential trouble spots very quickly. Since their initial experiment, there has been a major increase in the use of software metrics in industry; but, before describing it, it is worth outlining some history.



Ever since programming started, programmers have been asking: how good is my program design? Almost since this question was first posed, researchers have been trying to establish metrics, extracted from program designs and code, that could provide accurate answers.

False Dawn

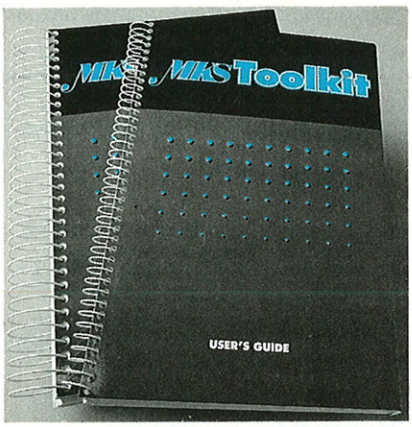
The first important attempt was by an American academic, Maurice Halstead, while based at Purdue University. Halstead believed that software systems obeyed some natural laws, in the same way that gases behave according to the laws of thermodynamics. Halstead said that if these laws existed, then there must be sets of equations which could be used to determine the quality of computer programs and systems.

In order to convince sceptics, and there were many of these, Halstead used a heady cocktail of computer science, information theory and thermodynamics to derive his equations. After four years' work, he published. Immediately, several experiments were launched in the United States. These experiments attempted to correlate factors such as the number of errors in a program, code readability and debugging effort against various metrics which had been derived from Halstead's equations and extracted automatically from the test code. Many early results seemed promising, and an explosion of US research effort followed. It was an unusual American conference that did not feature at least two or three papers reporting the results of yet another Halstead experiment. Millions of dollars were spent designing experiments and processing results.

It took two British workers, Peter Hamer and Gillian Frewin, both at the computer and electronics company STC, to draw at-

"The MKS Toolkit is an amazingly faithful replication of a System V UNIX™ environment." – UNIX Review

Essential UNIX Tools for any PC user



"The entire MKS documentation package proved excellent in every respect"

Daniel McAuliffe, IEEE Computer, Jan. 1990

We can tell you that MKS Toolkit offers both experts and novices the purest form of UNIX utilities in a DOS or OS/2 environment. Fortunately, we don't have to. Software reviewers, universities and major corporations all over the world are discovering MKS Software and how it can help their programmers and software developers make the most of their creative talents.

With our proprietary code, MKS Toolkit offers you more than 150 UNIX System V. Release 4 compatible tools for DOS or OS/2. With MKS Toolkit, your computer or clone becomes a comfortable environment for shells, string matching, editing, file manipulation, and more.

In this one package you will receive a complete:

- MKS Vi full – screen editor
- MKS Korn Shell – user programmable interface
- MKS AWK – programming and prototyping language
- profiler – to analyse the execution time of code
- egrep/fgrep – fast pattern matching
- archivers, sort/merge, encryption, data compression & much more

Over 150 commands that help you get the most productivity out of your PC.

Easy Installation

With MKS's menu-driven installation you can be up and running in less than 10 minutes, no matter what your level of computer expertise.

"MKS software is absolutely the best in its class. Don't mix environments without it."

Grover Righter, Novell

Interconnectivity & Lan Licences

MKS Toolkit interacts well on most standard PC and PS/2 networks. Combined with Novell Netware™, the most popular LAN for PC's, MKS Toolkit creates a UNIX time sharing system in DOS or OS/2 organizations. UNIX workgroups can now hook up all of their PCs using PC-NFS™ and MKS Toolkit, enabling you to use a PC as a UNIX workstation and off-load your mini or main machine.

Spanning Novice to Expert

This versatile software reflects it's users' needs. UNIX experts will have all the familiar power they demand while novices will find our documentation filled with excellent tutorial material.

Organizations such as AT&T, Hewlett-Packard, ITT, and NCR – all heavily committed to the UNIX system – use MKS Toolkit to create a standard operating environment. Universities from King's College to the University of Berlin use MKS Toolkit to enrich personal research computing environments and double the potential of their PC teaching labs.

"With this package, you can become familiar with the UNIX environment on your microcomputer, with DOS only a keystroke away."

Byte Magazine, May 1989

Cost-Effective Learning Tool

If your organization is committed to moving into the UNIX environment, then MKS Toolkit is the perfect learning path. DOS or OS/2 users retain the familiar world of their PC keyboard and programs yet can move to a desktop UNIX environment at a keystroke. Exposure to new commands and functionality now becomes an integral part of the novice's working day.

MKS Programming Platform

MKS Toolkit can be purchased as an individual product or you can buy MKS Programming Platform which contains MKS Toolkit and three other MKS products. MKS Platform helps smooth out the details of programming and software development by adding time-saving utilities such as: MKS RCS (Revision Control system), MKS Make (an automated program builder), MKS LEX & YACC (compiler learning and construction tools).

In all, you simply cannot find a more complete set of commands and utilities to get from DOS or OS/2 to UNIX. Once you have used MKS Toolkit, programming will never be the same again.

US List Price

MKS Toolkit (DOS & OS/2)	$399.00
MKS Toolkit (DOS/2 only)	$349.00
MKS Toolkit (DOS only)	$249.00

Please contact MKS dealers for UK domestic prices.

The Software Construction Co. Ltd. (0763) 73455; fax (0763) 73460
Grey Matter (0364) 53499
Systems Science (01) 833-1022; fax (01) 837-6411

MKS

35 King Street North, Waterloo, Ontario, Canada N2J 2W9

Put The Power Of UNIX In Your PC.

CIRCLE NO. 848

BEST BUY

"So simple to use you wonder why other packages make such a meal of resource allocation."

PC MAGAZINE

"InstaPlan offers the features of a £2,000 package with none of the trade offs of its low cost competitors."

PERSONAL COMPUTER EDITOR'S CHOICE

"We would recommend most warmly... InstaPlan as the best full featured project management package."

Practical Computing

(compared with PMW, Super Project and Harvard).

"InstaPlan is excellent, much better value than PMW, but keep improving it" Mr D Symington, CS Consultants

NEW InstaPlan 5000

InstaPlan 5000 is the latest addition to the InstaPlan family of project management software, which already has 25,000 users, from large multinationals to one man businesses, and is fully compatible with the rest of the InstaPlan family.

Easier Departmental Management

InstaPlan 5000 can gather individual project plans and consolidate them for departmental review, all with a single command on a stand alone machine or a PC network.

By bringing together the department's schedules and resource requirements in one model, you can recognise and resolve staffing conflicts and project priorities long before they become crises.

Most project management systems require all projects to be updated to the same date before merging. This is often impossible. InstaPlan 5000 allows projects to be combined freely.

Planning and Control for Larger Projects

InstaPlan 5000 will run on a standard 640K PC, Just like the rest of the InstaPlan family. But InstaPlan 5000 also supports EMS (Expanded Memory). This means that you can handle in excess of 10,000 Activities. Within these limits you are only constrained by the amount of EMS memory you have available. This is many times the capacity of some of InstaPlan's more expensive competitors.

Better Resource Management

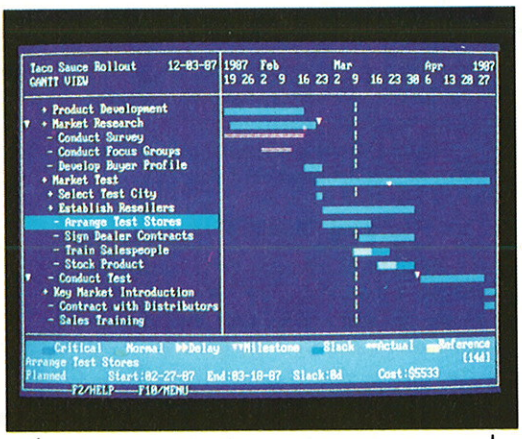
You can now level the load on a group of resources simultaneously, using he Adjust Loading command. The Locate Overload facility will scan all future periods for a resource, identifying any overloads.

We believe InstaPlan 5000 will provide the best solution for professional project managers and busy line managers who need project management to enhance their own and their department's performance.

See what InstaPlan users say:-

Easy to learn and Use

...it's design is just right. I set up a workload plan without even using the manual or any previous planning experience.
David Allen *Computer Services Manager*
Vallectric Ltd *Electrical Contractors*



With Quality Documentation

...a very good package, with excellent documentation.
Mr J Clarke *Finance Director*
Ceraf Ltd *Building Contractors*

...And Support too

...Good telephone support
Dr Robin Wilson
Harrogate Borough Council

Providing the Right Reports

...provides me with succinct and easy to understand documentation for Board presentations.
Dr R Gailey *Director of Research*
Tootal Thread Ltd

For Line Managers

Even though our busy line managers are only occasional users, InstaPlan gives them the tools to plan quickly and effectively.
Brian Alldridge *MIS Manager*
Prestwich Holdings plc

and for Project Managers

I am surprised to find myself learning from this product – an excellent investment.
Mr P M Cleary *Project Manager*
Maritime Dynamics Ltd

compared with the competition

Having suffered with Project Manager Workbench and SuperProject Expert, InstaPlan has provided us with the tools to produce all our reports and regular project updates easily. Management are impressed by the ease with which they can read and understand the reports.
Ms Petra Meyer *Product Manager*
Travicom Ltd

In summary

InstaPlan contains all the features and facilities required for project tasking, costing and tracking. An excellent, good value package, it forms the basis of successful project management within a business environment.
Andy Coster QA Manager,
Logsys

Money Back Guarantee

We are confident InstaPlan 5000 is the easiest project management system to learn and use, offering a significant price/performance advantage over all other systems. Buy InstaPlan 5000 now, if you return it within 30 days for whatever reason, we guarantee to refund your money by return.

Configuration: IBM XT AT PS/2 or compatible running under PC MS-DOS 2.1 or later with at least 500K of DOS memory and 3M of hard disk available to InstaPlan. Plan storage requires further DOS memory or access to EMS memory. EMS memory (LIM3.2+) supported to a maximum of 8Meg. A 5000 activity plan will require at least 2 Meg of expanded memory. InstaPlan 5000 will therefore run across networks using network shells like Novell where there is sufficient DOS memory available to InstaPlan. Printers: IBM ProPrinter & Graphics Printer, Epson FX/LX/LQ, Toshiba 351, HP LaserJet PLUS/Series II/ Deskjet. Input via keyboard, ASCII, ThinkTank or READY! Output to ASCII, Lotus or dBase. Charts are stored as bit mapped graphics and can be printed but not viewed on screen. ALL MARKS ACKNOWLEDGED. DEALER ENQUIRIES WELCOME. QUANTITY DISCOUNTS AND SITE LICENCES AVAILABLE. TRAINING AND CONSULTANCY AVAILABLE IF REQUIRED.

DEEPAK SAREEN ASSOCIATES

Freepost, 28 Preston Rd, Wembley HA9 8BR
Phone 01 908 2423 Fax 01 904 6668

InstaPlan 5000 £499 plus £6 p&p and VAT

☐ **Please send me InstaPlan 5000.**

☐ I enclose a cheque for £580.75

☐ Please bill my Visa/Access Card

☐ **send me more details of InstaPlan 5000 and the InstaPlan family, from only £99.**

Company .....

Name.....

Signature .....

Phone no. .....

Please enclose a compliments slip showing your required delivery address and credit card details if appropriate.

SM 3/90

CIRCLE NO. 849

tention to the major problems with Halstead's work. In a paper that stunned the metrics world, they pointed out that the Halstead work had very little theoretical validity and, moreover, the experiments that had been performed were almost invariably badly designed, and used poor statistical methods.

Frewin and Hamer's work nearly ended interest in metrics as a useful tool. Some original research done in the mid-seventies saved the day. The work was carried out by Thomas McCabe, a member of staff at the American National Bureau of Standards. McCabe was curious how the structure of programs made certain tasks more difficult. In particular, he was interested that complex control flows in a program made it difficult to test and understand.

Graphic Approach

In order to explore this phenomenon McCabe modelled a program as a mathematical structure known as a directed graph. This consists of a series of nodes and connecting lines known as arcs. A node represents a decision or the end of some processing in a design or program, while the arcs represent some processing that occurs.

A simple program fragment is shown in Figure 1, its directed graph appears in Figure 2. Nodes are labelled with lower case letters, arcs with numbers. The arrow attached to an arc shows the direction of processing. The figure shows that the fragment can be represented by two nodes: *a* and *b*. Node *a* represents the decision in the `IF` statement and node *b* represents the point where the processing in the two branches of the `IF` statement join up.

If one assumes that a leftmost arc represents the true branch of the `IF` statement, and the rightmost arc represents the false branch, then arc 1 represents the processing `errorcount:=errorcount+1` and arc 2 represents the processing `normalcount:=normalcount+1`.

Figure 3 shows a slightly more complex example, Figure 4 is its directed graph. It consists of five nodes and seven arcs. Node *a* represents the decision in the `WHILE` statement. Node *b* is the decision in the first

```
IF temp ø 200 THEN
  errorcount:= errorcount +1
ELSE
  normalcount:= normalcount+1
```

Figure 1 - Simple Code Fragment

`IF` statement, node *c* represents the destination of both parts of the processing associated with the `IF` statement, and so on. Similarly, arc 1 is associated with the processing `Read(data)`, arc 2 with `Print('error in data')`, arc 3 with `sum:=sum+data`, arc 4 with the loop back to node *a* to re-execute the `WHILE` statement, arc 5 the transfer of control from the `WHILE` statement to the second `IF` statement and, finally, arcs 6 and 7 represent the two actions `sum:=0` and `average:=sum/50`.

Halstead used a heady cocktail of computer science, information theory and thermodynamics to derive his equations

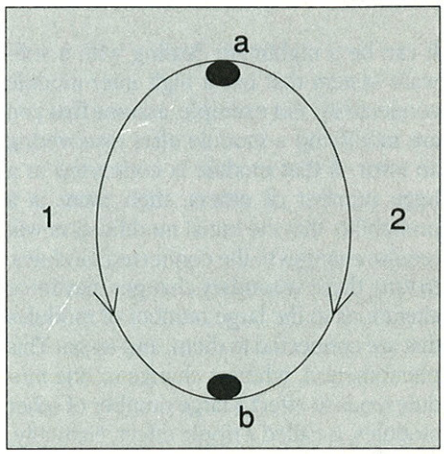
McCabe claimed that a metric used in the mathematical study of directed graphs gave a very good indicator of code quality. This metric is known as cyclomatic complexity and it is defined as:

`no. of arcs - no. of nodes + 2`

The more unstructured a code module is, the higher is its cyclomatic complexity. If we apply this definition to the graph shown in Figure 4, 7 - 5 + 2 = 4 we obtain a result of '4' for the fragment shown in Figure 3. This rating suggests that the code is well structured. As a rule of thumb, although the technique is slightly language-dependent, modules scoring a value higher than 20 are considered poorly structured, and should be regarded as trouble spots. The software world now recognises McCabe's work as successful. Cyclomatic complexity is widely employed in industry to judge the degree of 'unstructuredness' in a module.

There are drawbacks. One of these is that cyclomatic complexity can only be measured late in a software project after a system has been specified, designed and programmed. Information from this metric is, therefore, only of limited help to the beleaguered project manager. By the time his project reaches the stage that he can use

Figure 2 - Directed Graph of Figure 1



the metric, he has no resources left to recover from poor structure that it reports. Because of this, researchers have, since the late 70s, been looking for a software metric that somehow quantified the quality of a system early in the project as early as system design.

During the 1980s, a number of groups devised metrics which looked very promising. Remarkably, their work sprung from concepts originally suggested two decades before. In particular, it was based on ideas of one Christopher Alexander, an Englishman who worked at Harvard University as an architect.

Alexander, who is regarded as a minor guru of design, wrote a seminal work on the subject called *Notes on the Synthesis of Form*. This book - an exceptionally difficult one to read - discusses good and bad design, and describes the principles behind the best design practices.

Buried in the middle of the book is a description of the 'well- designed artefact': it should be designed in such a way that each component is as isolated from the others as possible. For a piece of electronic equipment the principle is clear - a system should be highly modular, so that it can be easily maintained. An error in a well-designed circuit is simply rectified by removing the circuit board and replacing it with another. Provided there are no complicated connections, this can be carried out easily. So what does Alexander's principle mean when applied to software?

The answer is that a good system design is one in which the modules are in relative isolation to each other. The consequence, in technical terms, is that the connections between modules should be minimised. Modules should not have large numbers of parameters, they should not call large num-

bers of other modules and neither should they not reference too many shared global variables.

It can be a nightmare dealing with a software system that has a high inter-module connectivity. For example, assume that you are modifying a module after discovering an error. If that module is connected to a large number of others, then there is a probability that the initial modification will require changes to the connected modules. In turn, these secondary changes may force alterations to the large number of modules that are connected to them, and so on. This phenomenon, where a change to one module tends to affect a large number of other modules, is called a ripple effect. Naturally, this effect can cause massive resource headaches for the project manager.

Since 1980, various universities and research laboratories have developed several metrics based on connectivity. The British centres of excellence in this type of metrics research are ICL, British Telecom, the Open University, Wolverhampton Polytechnic, City University and the National Computer Centre in Manchester.

Most of these metrics can be extracted early in the software project, just after system design has been completed, and are based on a count of parameters, number of modules, global variables and calls on other modules. The major aim of these metrics is to devise a structure known as an impact matrix.

An example of such a matrix is shown in Figure 5. It shows the impact matrix for a system which contains five modules named A-E. The entries in the matrix represent probabilities. Each value is the probability that a change in module *row* will produce a change in module *column*. For example, the probability that a change in module A cause a change in module B is 0.2. A change occurring in a given unit will always cause

```
WHILE data 0 DO
     Read(data);
     IF data 100
     THEN
    Print('error in data')
     ELSE
          sum:=sum+data
     END_IF;
END_WHILE;
IF error_flag THEN
     sum:=0
ELSE
     average:= sum/50
END_IF;
```

Figure 3 - More Complex Code Fragment

an alteration to that same unit, so the leading diagonal of the matrix will always be filled with 1. Such a matrix enables the project manager to gauge the effect of change on the system that he is developing.

Metrics in Use

Over the last five years software metrics have moved out of the laboratory into the software industry - notably in the United

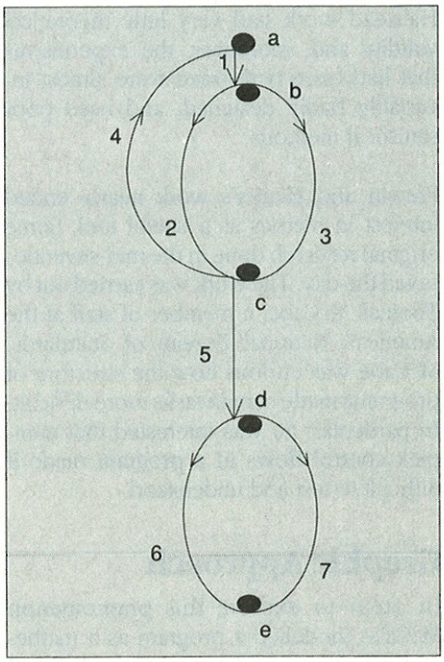
Poorly made changes have resulted in structure degrading - the software equivalent of rusting

States and Japan. There are a number of uses for such metrics. One of the most important is in software maintenance: the process of modifying a system once it is in operation.

It may come as a surprise that maintenance is a major expense for software companies. In 1980, a large survey of American commercial data processing companies looked at the software practices of over 450 developers. From this came the result that, on average, over 70% of the companies' software budgets were spent on software maintenance.

When you look at the types of change that a system undergoes when it is in service, the reason for this becomes clear. First, there are corrective changes. These occur when the customer detects errors committed during development, and the developer has to put them right. Second, there are perfective changes, where the completed system is improved - often by speeding it up. Third, there are adaptive changes which arise from shifts in the user requirements. For example, an accounting system may have to be changed when new tax laws are announced, or a defence system may need to be modified when a potential aggressor changes his tactics or technology. It is this last category of change

Figure 4 - Directed Graph of Figure 3



that tends to be the largest consumer of resource. Changes in requirements for large systems come in thick and fast, particularly if the system is successful.

The project manager in charge of maintenance may experience a strange phenomenon. A change to his system during its early life may consume a small amount of resource, but a year or so later the amount of resource required for a similar change is larger; in three or four years the amount of resource has increased by two orders of magnitude.

This is what has happened. His system has gone out of control: poorly made changes have resulted in the system structure degrading - the software equivalent of rusting. This process is rapid and has an exponential character; on a very rusty system massive amounts of resource are devoted to achieve just a few changes. What eventually happens is that the system becomes unmaintainable and the code is not altered. This was the fate of the rogue module in the UNIX system, described at the beginning of the article. Unalterable modules become less and less useful and are eventually retired.

Software metrics, and in particular system design metrics, help the project manager to monitor this system rusting. With the extra information he can choose this moment to intervene and send in designers and programmers to clean up the system architecture. The intervention can arrest the degradation and keep the system running much longer. Figure 6 shows the rise in

complexity of two systems. One system has been left to degrade, periodic changes to system structure have been applied to the other. Often these changes require low levels of resource. Several studies of system architecture, which have used metrics, have suggested that only a small proportion of modules in a system - frequently less than 10% - are badly affected by system rusting.

The other area where metrics are useful is in software project standards. One useful standard is to specify that, if a completed module has a cyclomatic complexity greater than a particular number, that module must be redesigned. Subsequent unit testing and integration testing will be much less error- prone than normally.

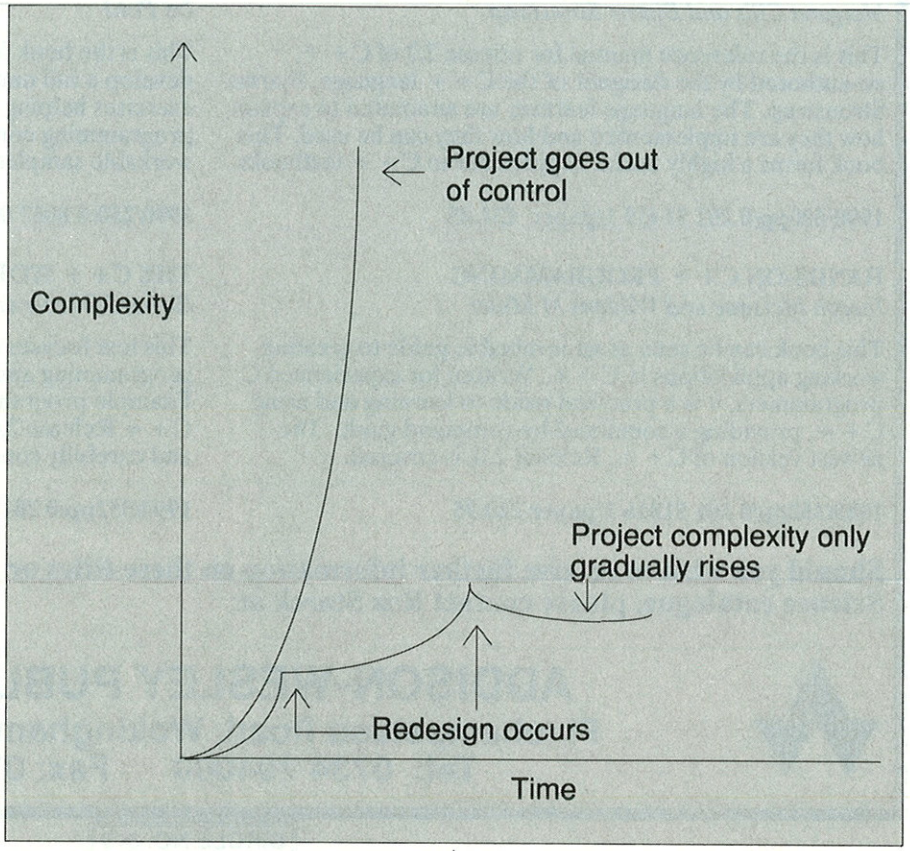
Metrics are now being used in many companies: in America in organisations such as Boeing, Computer Sciences Corporation and NASA; in Japan in companies such as Hitachi and NEC; and in this country in British Telecom, ICL, STC, the National Westminster Bank, GEC/Plessey Telecommunciations and the software company Data Logic. The extent to which metrics are used in industry often surprises me. Five years ago, when many researchers, even those ardently attracted to metrics, would have said there was little industrial penetration, I addressed an organisation called the Software Quality Assurance Forum. This is an admirable collection of the quality assurance departments of major British companies who periodically meet to discuss industry trends, swap experiences and listen to external speakers. I was one of the speakers, and the topic I was discussing was metrics.

I find it very difficult talking to industrial audiences. I am normally introduced as Professor Ince or as Doctor Ince. Immediately I see a reaction on the faces of my potential listeners, which suggests that, while they will listen to what I say with great interest, they do not expect to hear anything from an ivory tower dweller such as myself that will be applicable within the next two millennia. My tactic on this particular occasion was to assert, in the very first sentence of my talk, that the vast majority of my audience used software metrics.

	A	B	C	D	E
A	1	.2	.4	.1	.3
B	.1	1	.6	.1	.2
C	.1	.2	1	.1	.3
D	.2	.2	.1	1	.2
E	.1	.1	.1	.1	1

Figure 5 - Example Impact Matrix

Figure 6 - Variation in complexity against time, for two systems.



Each member of my audience reacted by glancing sideways at his neighbour, rather as though I had made an obscene remark.

They don't expect to hear anything from an ivory tower dweller that will be applicable within the next two millennia

I ploughed on. I said that I was sure that they all used the metric where their programmers had to test modules with data that executed every single statement in the code. Then I made my big blunder: I asked everybody who used this metric to put up his hand. Just three hands went up, out of an audience of over 100. It was the worst beginning to a talk that I have ever given.

However, in the afternoon, I joined one of the workshops. I discovered that most participants in the workshop were actively using metrics. In particular, they were using a metric known as a function point count, which is used to predict the amount of resources required on a project. At that time, this metric had received only a little validation; they were, in metrics terms, living in the fast lane. From the evidence of senior software staff, large numbers of companies are now putting software metrics at the top of the agenda for innovation. These days, everyone's going metric.

.EXE

Darrel Ince is a Professor of Computing Science at the Open University, where he is the acting head of the computing department.

If you are interested in reading further, Professor Ince recommends his own book Software Development: Fashioning the Baroque (Oxford University Press), which contains an introduction to metrics, and Controlling Software Projects (by T DeMarco, published by Yourdon Press, ISBN 0-917072-32-4) for more thorough coverage.

The Evaluator

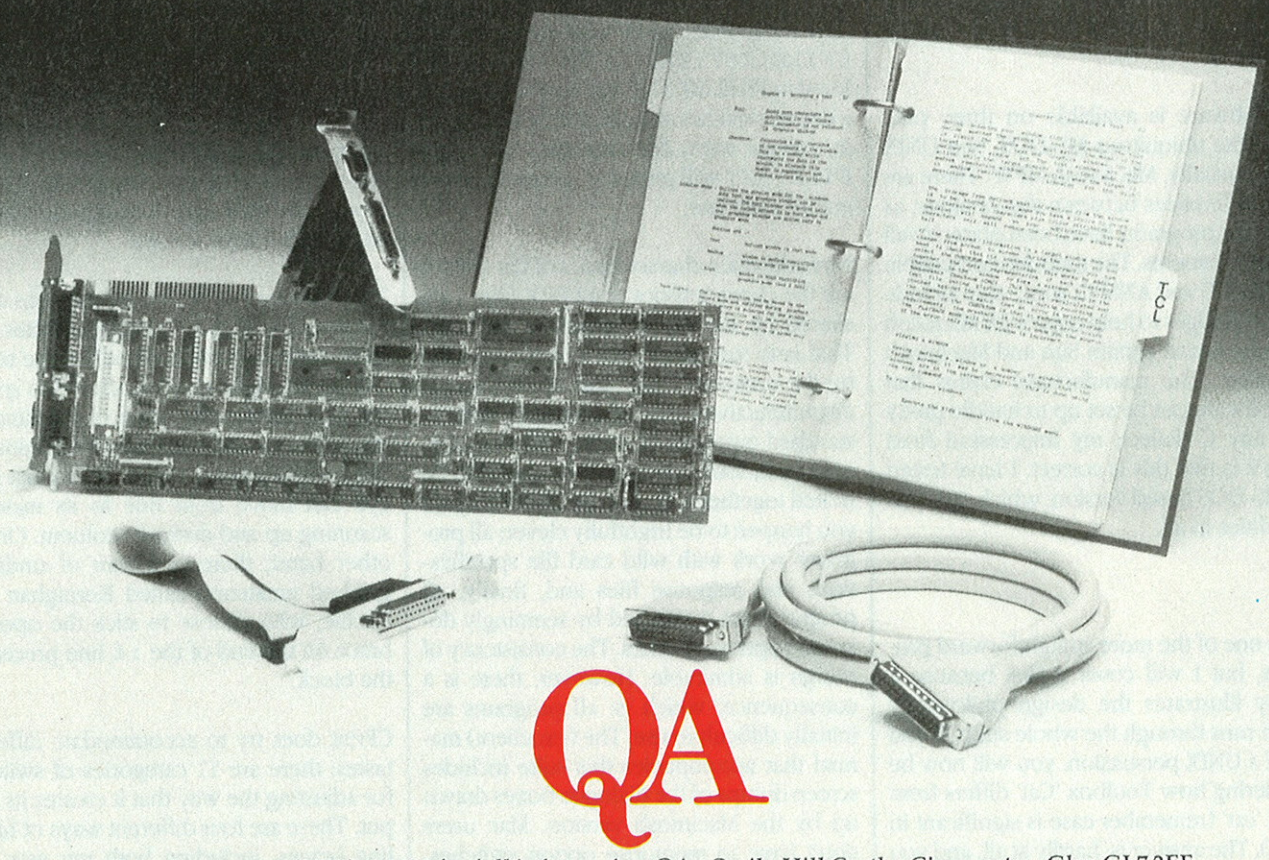
The Breakthrough in Software Testing

Completing the writing of your software is half the battle.

Ensuring that you have produced reliable fault free code is yet another major challenge. Testing software to be certain that full functionality is met during development or maintenance cycles is costly, time-consuming and prone to error...until now.

Evaluator, created by Elverex and supplied by QA, both simplifies and automates software testing, allowing you to create a comprehensive test process which fully exercises software to detect any errors. It saves effort, repetitive work, and frees the time of test personnel.

- *Interactive recording and playback of complex test sequences*
- *Rich test control language to write test programs*
- *Totally non-intrusive testing of any application on any operating system – DOS, Unix, OS/2, VMS, RMX...*
- *Automatic test report generation*
- *Unattended operation allows for more productive use of test personnel*
- *Easy to use control software for test management and monitoring*
- *Uses standard IBM PC or AT or 100% compatible as test station*
- *Tests software that runs on terminals or on a PC*



QA

Telephone us on (0285) 655888 for full information. QA , Cecily Hill Castle, Cirencester, Glos GL7 2EF

CIRCLE NO. 851

Without tools he is nothing

The C Programmer's Toolbox is a set of software utilities that you didn't know you needed. We asked Will Watts to have a rummage in it.

You have the C compiler. You have the editor. You have the linker, the librarian, the symbolic debugger and the MAKE utility. I bet you think that you are all set to start work. MMC AD Systems, producers of the *C Programmer's Toolbox*, would have you believe otherwise. This package consists of a pot-pourri of 24 utilities, ranging from the very complex, supporting their own programming languages, to trivial little programs of the kind that you might knock out for yourself in a few minutes - albeit in cruder form - to meet a particular need. Not all of them are directly C-oriented; some are just slightly obscure general-purpose utilities, like the kind that one finds lurking in the murky depths of the UNIX operating system.

The software is available on three platforms: the ubiquitous MS-DOS, Sun UNIX and, unusually, Macintosh MPW. There are slight differences between the versions, as not all the programs have been ported to all the environments. The tools are compatible with K&R C and ANSI C; some also include special kludges to help cope with Microsoft C, Turbo C and certain Sun and Macintosh compilers. The manufacturer claims that the software can be set up to handle pretty well any C dialect; my impression from using it is that this is correct. I have tested the MS-DOS based version, which contains 21 utilities in all.

Cat

Cat is one of the more straightforward programs, but I will cover it first because it clearly illustrates the design philosophy which runs through the whole suite. If you are of a UNIX persuasion, you will now be wondering how Toolbox 'Cat' differs from UNIX 'cat' (remember case is significant in UNIX). The answer is: hardly at all, and you may now leave the class-room and frolic in the sunshine until I get to the next utility. Everybody else stay here, pencils at the ready.

Cat is very similar to MS-DOS's TYPE command. If you type `Cat AUTOEXEC.BAT` at the DOS prompt then, sure enough, AUTOEXEC.BAT goes scrolling past. Cat writes to the standard output, so you can store its output in a file. This feature can be used to perform concatenation: `Cat AUTOEXEC.BAT CONFIG.SYS >MESS.POO`. Cat understands wild card file specifications (ie unlike TYPE, `Cat *.BAT` works) and indirect file specifications, which work rather like the response files sometimes used with linkers: `Cat @FILELIST.TXT` TYPEs all the files listed in FILELIST.TXT. Cat also accepts half a dozen switches, which modify the output in various ways; for example, `Cat /ev FILE.TXT` will print a '$' character at the end of each line.

Now, the main characteristics of Cat - which are the characteristics of a UNIX filter - are shared by *all* the programs in the suite. That is to say: all programs are controlled by the command line, and do not accept any interactive input; all programs write to standard output (and some, including Cat, read from standard input), so they can be bolted together in frightfully clever ways, if you happen to be frightfully clever; all programs work with wild card file specifications and response files and, finally, all programs are controlled by seemingly dozens of option switches. The consistency of design is admirable. However, there is a consequence, which is: all programs are initially difficult to use. The (excellent) manual that accompanies this suite includes screen dumps of the dialogue boxes drawn up by the Macintosh version. Mac users don't have to memorise option switches, they just select labelled check boxes. After playing with a few of these programs, I wished that I was a Mac user.

Is Cat useful? Well, I have not found a use for it; but I harbour the suspicion that, if you kept it around, you would come across the task for which it was written. There are quite a few little programs like Cat in this package, but perhaps it is time to deal with the biggies.

Oh so pretty

The CPrint program 'reformats and beautifies C source files', ie it is what is generally known as a 'pretty printer'. This utility accepts (compilable) C code as input, and spews it out exquisitely rearranged; every line of code standing to attention at the correct indentation, each function argument parked exactly one respectful space away from the comma of its predecessor, as though the stuff had been typed in by the angels. This is the theory.

The drawback is that everybody has a different opinion as to what constitutes perfectly formatted code. Myself, I like to see it done the **right** way, which is two spaces for each indent, the name of the function repeated as a comment after its terminating brace, and all braces lined up, so that your eye can move from one to its mate by scanning up and down its column. On the other hand, there is a pair of undistinguished amateurs, called Kernighan and Ritchie, who choose to stick the opening brace on the end of the `if` line preceding the block.

CPrint does try to accommodate different tastes: there are 11 categories of switches for adjusting the way that it creates its output. There are four different ways of handling braces, including both my way and K&R's. There is an option, included for the

Figure 1 - Untidy code, prior to CPrint

```
void move_points() /*Move all active points throughlstep*/
{inti,j;
num_active=0;    num_dormant=0;
    for (j=0;j<=scan;j++)
        if (alive[j]<=0) {
            if (j<sep) for (i=1;i<=slowdown;i++);    if (alive[j]<0) num_dormant++;
        } else {
            if (noise>0) if (rand()<noise) outportb(0x61,3);
            if (state[j]>0) show(x[j],y[j],col[j],size[j]);
            else if (state[j]==0) state[j]=1;
            num_active++;    alive[j]--;    yvel[j]+=g[j];    x[j]+=xvel[j];
            if ((y[j]+=yvel[j])>ymax) {y[j]-=yvel[j]; xvel[j]=0;}
            if (alive[j]==0)
                if (g[j]==0) {
                    g[j]=gv;    alive[j]=10;
                } else if (size[j]>1) {
                    size[j]--;    alive[j]=fade[j];
                } else {
                    col[j]=0;    if (j<sep) {next[j]=freep;    freep=j;}    }
     show(x[j],y[j],col[j],size[j]);    outportb(0x61,0);}
    while ((scan>=sep) && (alive[scan]<=0)) scan--;
    if ((noise=abs(noise))>0) noise-=300;} /*End move_points*/
```

benefit of users of the Brief editor, that lets you format function definitions so that the function name starts in column 1:

```
double
MathsFunction()
    {
    int test;
    ...
    }
```

However, all this is not enough. The first important omission is the lack of control that CPrint gives you over indentation. Essentially, you can have either four spaces for each indent or, if you like living dangerously, one tab character. There **is** a pair of utilities in the toolkit, called EnTab and DeTab, which allows you to convert white space in text files to and from Tab characters. Since these programs allow you to set tab stops, it is possible to reformat CPrint's output 'tabbed' code with arbitrary indentation - the manual invites you do to just this. However, changing the indentation this way destroys the right hand margin justification of comments that is, for me, the main reason for using CPrint. Without getting entangled in further detail, you may take my word for it that the whole business of indentation control is a dog's breakfast.

Further complaints: CPrint does not perform a complete parse of the input code like a compiler; in particular it does not understand preprocessing directives. This is the sort of error you get:

```
if (inst->saved == NULL)
#ifdef __TURBOC__
  lineto(x,y);
#else
/* Not indented */
while (x > 0)
...
#endif
```

A final problem, true of any pretty printer but seemingly more so of this one, is most easily described as 'lack of common sense'. This applies particularly to the formatting of comments. For example, it is my habit, as I mentioned above, to end each function like this:

```
} /* MyFunction */
```

CPrint inserts two linefeeds between the comment and the brace, so that it ends up, stranded, half-way between one function and the next.

Perhaps I am harsh on CPrint. I just found it insufficiently flexible for my needs (I would have liked to have used it to force all C code submitted for publication in .EXE into house style). Figures 1 and 2 contain an extended before/after example, using a function from a Turbo C program called PYRO, by KG Shields, so you can make up your own mind. Whatever else, I dare say project managers will find it a useful tool for stamping out tiresome individualism among programmers.

Power of Three

CFlow, PMon and CritPath are three distinct utilities, which interact in an ingenious way to tell you where (if not how) your program needs rewriting. To explain from the beginning: CFlow analyses your source code for the control flow between functions. It produces five reports, including a list of library calls (ie calls to functions not defined in the code), an alphabetical list of functions with 'called by' information and a program flow tree. The latter is equivalent to a hierarchical structure chart, printed using outliner

```
void move_points()        /*Move all active points through 1 step    */
    {
    int i, j;
    num_active = 0;
    num_dormant = 0;
    for (j = 0; j <= scan; j++)
        if (alive[j] <= 0)
            {
            if (j < sep)
                for (i = 1; i <= slowdown; i++)
                    ;
            if (alive[j] < 0)
                num_dormant++;
            }
    else
        {
        if (noise > 0)
            if (rand() < noise)
                outportb(0x61, 3);
            if (state[j] > 0)
                show(x[j], y[j], col[j], size[j]);
            else if (state[j] == 0)
                state[j] = 1;
            num_active++;
        alive[j]--;
        yvel[j] += g[j];
        x[j] += xvel[j];
        if ((y[j] += yvel[j]) > ymax)
            {
            y[j] -= yvel[j];
            xvel[j] = 0;
            }
        if (alive[j] == 0)
            if (g[j] == 0)
                {
                g[j] = gv;
                alive[j] = 10;
                }
            else if (size[j] > 1)
                {
                size[j]--;
                alive[j] = fade[j];
                }
            else
                {
                col[j] = 0;
                if (j < sep)
                    {
                    next[j] = freep;
                    freep = j;
                    }
                }
            show(x[j], y[j], col[j], size[j]);
            outportb(0x61, 0);
            }
        while ((scan >= sep) && (alive[scan] <= 0))
            scan--;
        if ((noise = abs(noise)) > 0)
            noise -= 300;
        }


/*End move_points                                                    */
```

Figure 2 - Code as processed by CPrint

S-GKS portable graphics toolkit

- available on a wide range of systems from PC's to mainframes
- support for the popular workstations from SUN, Apollo, IBM, HP, Sony, DEC
- Fortran and 'C' language support
- X-Windows driver
- CGM (Computer Graphics Metafile) support
- Training courses run for GKS programmers

B-GUL graphics toolkit

- Business & technical graphics toolkit layered on top of GKS for efficiency and portability
- Comprehensive charting facilities including;
  Bar-charts, Pie-charts, XY-plots
  Word-charts, Tables, Keys, Hi-lo-charts
- Pasteboard facility for composite plots
- Powerful data scaling & axis facilities
- FORTRAN and 'C' language support

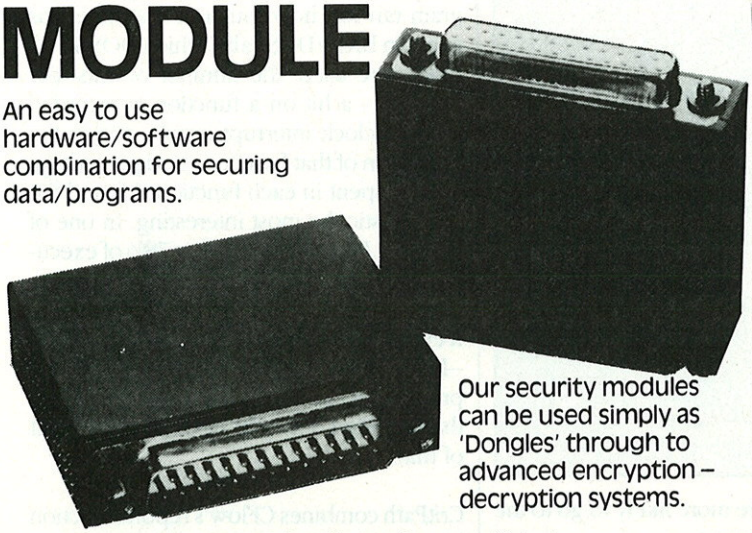
SSL

Scientific Software Limited
Rose Industrial Estate
54 Marlow Bottom
Marlow, Bucks SL7 3ND
Tel : 0628 890011

CIRCLE NO. 856

SOFTWARE SECURITY MODULE

An easy to use hardware/software combination for securing data/programs.



Our security modules can be used simply as 'Dongles' through to advanced encryption – decryption systems.

- ★ For IBM PC family (and compatibles)
- ★ Variants for other systems
- ★ Advanced construction – rugged and compact
- ★ Proven throughout the world
- ★ Transparent operation – user friendly
- ★ Driver software supplied for variety of applications

CTL

Control Telemetry of London
11 Canfield Place, London NW6 3BT
Tel: 071 -328 115 Fax: 071 -328 9149

CIRCLE NO. 857

System Science

C COMPILERS			
Microsoft C ver 5.1	£235.00	Turbo C	£75.00
Microsoft Quick C2	£65.00	Turbo C professional	£155.00
JPI Topspeed C-std	£149.00	Watcom C ver 7.00	£235.00
JPI Topspeed C-ext	£295.00	Lattice C (new ver 6.0)	£185.00
Zortech C++ Ver 2.0	£125.00	Glockenspiel Common View	£375.00
Zortech C++ Prof Der	£275.00	Quick C with Quick ASM	£129.00
C GRAPHICS			
Halo-88 (also Qb, Fort, Pas)	£195.00	Enhanced Graphics T'kit	£275.00
Metawindows-no royalties	£185.00	Metawindows Plus	£245.00
Ingraf-C-source, no royalties	£175.00	PCX effects	£85.00
PCX programmers Toolkit	£165.00	MS Windows SDK	£295.00
C DATAFILE			
Btrieve (also QB, Pas, Fort etc)	£169.00	Ctree — source	£255.00
Btrieve/N -network	£395.00	Fabs Plus	£125.00
XQL addon for Btrieve	£525.00	Code BASE 4	£195.00
Xtrieve +	£395.00	CQL for Ctree or Btrieve	£295.00
C SCREENS			
Greenleaf Data Windows	£195.00	Aspen Cubes — source	£225.00
Panel Plus II	£275.00	Formation Source	£265.00
C Worthy	£call	Cscape with Look & Feel	£315.00
C COMMS AND OTHERS			
Greenleaf Comms	£175.00	Essential Comms s'rce	£245.00
Blaise C Asynch Manager	£125.00	C Lisp — source	£295.00
Blaise Turbo C Tools- new ver	£95.00	Blaise C Tools Plus	£95.00
Greenleaf Functions	£125.00	Greenleaf Super Funcs	£165.00
PROGRAMMERS TOOLS			
Microsoft Assembler 5.1	£90.00	Turbo Assembler/Debugger	£99.00
Brief Editor (DOS or OS/2)	£145.00	Epsilon Editor (DOS/OS/2)	£145.00
C Toolbox from MMC	£165.00	Vedit Plus new ver	£99.00
Plink 86 Plus	£395.00	RT Link	£165.00
PolyMake	£115.00	PVCS Corporate	£375.00
MKS Toolkit	£155.00	Vq matchmaker & editor	£175.00
CROSS DEVELOPMENT			
2500AD Cross Assemblers	£135.00	2500AD Simulators	£115.00
Avocet Cross ASM	£295.00	Avocet Simulators	£395.00
IAR Cross Assemblers	£185.00	Avocet C Cross	£695.00
IAR C Cross Z80, 6301 etc	£995.00	Introl C Cr. 6809, 6301, 68000	£1445.00
Aztec C Cross	£275.00	LMI Forth Metacompiler	£495.00
FORTRAN ETC			
Microsoft Fortran with CodeView	£235.00	RM Fortran with Forte	£465.00
Lahey F77L (new version)	£395.00	Halo 88	£195.00
Lahey F77/32EM for 386s (new version)	£795.00	Ingraf Fortran source	£175.00
FTN 7/386 Salford	£695.00	Enhanced Graphics T'kit s'rce	£175.00
GSS GKS Graphics	£385.00	FortLib source	£95.00
PASCAL ETC			
Turbo Pascal 5.5	£75.00	Blaise Power Tools Plus	£99.00
Turbo Pascal Professional	£155.00	Blaise Turbo Asynch	£125.00
Quick Pascal	£60.00	Microsoft Pascal 4.0	£165.00
Fabs Plus Datafile	£125.00	Many more for Turbo Pascal	£call
BASIC ETC			
Microsoft QuickBasic ver 4.5	£65.00	Quick Comm	£95.00
Microsoft Basic 7 (new)	£275.00	Crecent Quick Pak Prof	£115.00
Btrieve	£169.00	Fabs Plus Datafile	£125.00
Blaise Power Screen	£95.00	Many more	£call
OTHER LANGUAGES			
JPI Modula-2	£59.00	Logitech Modula-2 Develop	£175.00
MuLisp 87 Interpreter	£175.00	MuLisp-87 Compiler & interp	£225.00
Smalltalk V	£79.00	Smalltalk 286	£129.00
PC Forth by LMI	£95.00	UR/Forth by LMI	£225.00
Microsoft Cobol II	£445.00	PC Logo	£75.00
UTILITIES & OTHERS			
Uniform/PC read CP/M disks	£62.00	Dan Bricklins Demo II	£175.00
Spin Rite II	£69.00	Page Garden	£85.00
Unidos with 8MHz Z80	£125.00	Sourcer with BIOS	£115.00
Q Ram (new)	£55.00	Vfeature Deluxe	£85.00
dBrief addon for Brief	£79.00	Tom Rettigs Library	£75.00
Above disc Lim Dr	£69.00	Math CAD 2.5	£345.00
Microstat II	£295.00	Derive — computer Algebra	£115.00
386 AND OS/2			
385Max Prof	£95.00	VM/386	£115.00
DesqView 386 (new version)	£125.00	Xenix 386 — new price list	£call
OS/2 1.1 Stand Ed (IBM)	£call	OS/2 PM Toolskit	£295.00
Many OS/2 Libs & utils	£call	Many 386 Utils	£call
QEMM (new)	£60.00	Win Pro/PM	£250.00
Smalltalk/PM	£325.00	Multiscope Debugger	£195.00

All prices are exclusive of VAT but do include UK delivery.

A sample only of our extensive range. Please enquire for unlisted items

Access

Prices are subject to change — Please confirm

3-5 Cynthia Street, London N1 9JF
Tel: (01) 833-1022,
Fax: (01) 837-6411

BARCLAYCARD VISA

CIRCLE NO. 858

Figure 3 - Simple CFlow output

```
            *** Raw Flow Report ***
            -----------------------


Processing file: SDB.C


InitB(26).
CleanUpB(31).
SetPattern(35)=>>setfillstyle(42)-bar(43)-putpixel(47)-putpixel(48).
SetScenario(194)=>>memcpy(228)-CreateE(251)-CreateG(255)-Tunnel(261)-
    SetNobStart(264).
....


            *** Master Defined Function List ***
            ------------------------------------


AddObject(SDU.C:76).
Background(SDB.C:270)=>free(2*)-getimage(2*)-imagesize(2*)-malloc(2*)-
    putimage(2*)-SetPattern(1)-SetScenario(1).
BitToDirection(SDD2.C:112).
BroadcastList(SDM.C:33).
Calibrate(SDM.C:51).


...


            *** Undefined Function List ***
            -------------------------------


abort() bar() closegraph() cprintf() detectgraph() fprintf() free() getch()
getimage() getmaxcolor() getvect() imagesize() initgraph() kbhit() line()
malloc() memcpy() putimage() putpixel() registerfarbgidriver() setcolor()
setfillstyle() setvect() __cli__() __outportb__() __sti__()



            *** Program Flow Tree ***
            -------------------------


  1:    main(SDM.C:139)
  2:        Background(SDB.C:270)
  3:            free(*)
  4:            getimage(*)
  5:            imagesize(*)
  6:            malloc(*)
  7:            putimage(*)
  8:            SetPattern(SDB.C:35)
  9:                bar(*)
 10:                putpixel(*)
 11:                setfillstyle(*)


...


            *** Function Called by List ***
            -------------------------------


abort(*) <= CutObject(1).
AddObject(SDU.C:76) <= MakeDigger(1)-MakeNobbin(1).
Background(SDB.C:270) <= main(1).
bar(*) <= SetPattern(1)-Tunnel(2).
BitToDirection(SDD2.C:112) <= SetDirection(2).
BroadcastList(SDM.C:33) <= ProcessQueue(3).
Calibrate(SDM.C:51).
```

conventions (lower level functions are indented).

Although the program flow tree is not as immediately readable as a graphical structure chart, as produced by a utility such as *Clear+ for C* (reviewed in the September '89 issue of .EXE Magazine), I think CFlow's text-only printed report is much more practical. Apart from anything else, it takes a fraction of the time to print out - which means that you are more likely to go to the trouble of producing it.

Like Clear+, and presumably all source code analysis programs for the foreseeable future - it will be an extremely smart piece of coding that conquers this one - CFlow is flummoxed by calls to dereferenced pointers to functions. CFlow records the call, but it has no idea where you are going. Functions that are accessed only indirectly appear never to be invoked. Since calling pointers is one of the most powerful and much-used features of the C language, this is an important limitation. It also produces a knock-on effect in the CritPath utility.

CFlow also generates a Raw Flow Report, which lists each defined function, together with the routines that it calls - this data is passed on to CritPath. Samples of output from CFlow are shown in Figure 3.

PMon is an execution profiler. This is how it works. You take your application's .MAP file, as generated by the linker (Microsoft's was expected, but Borland's TLINK was sufficiently compatible) and pass it through a filter program called MapVar. You should now have a file listing the addresses of all the public labels (`static` functions, you notice, confuse the issue). Next, you start up PMon, with the name of your .EXE file as an argument. PMon sucks in the data from your .MAP file, and installs intercepts on the clock interrupt and also on certain (user-specifiable) DOS and BIOS functions. At this stage it occupies about 30 KB of memory. Next, it loads your application above itself and executes it. Every BIOS and MS-DOS call made by your program is recorded. At each clock interrupt, PMon registers the current position of the program counter. When your application completes, PMon tabulates the information and writes it out to a file.

The PMon output file contains all sorts of information. It tells you how long your program ran for, how much of that time was spent in BIOS/DOS calls, which DOS/BIOS calls were used, the number of 'hits' per function - a hit on a function is recorded when a clock interrupt occurs during the execution of that function - and percentage of time spent in each function. I found the last statistic the most interesting. In one of my tests, PMon reported that 78% of execution time was spent in just one function (no, clever clogs, it wasn't the keyboard loop). It even printed the message: 'HINT---HINT--HINT Concentrate on this function to improve your program's performance.' Why, thanks PMon. I should never have thought of that.

CritPath combines CFlow's report (function hierarchy) with PMon's report (time spent in each function) to try to determine the application's critical path, ie the sequence of functions, called from `main()`, that consumes more execution time than any other sequence. This is more tricky than you might, at first, think. For instance, if one function is called by two different functions, which should be charged for the time used by the called function. The problem

Figure 4 - Weighted Hierarchical Program Flow Tree

```
        Act Rank   Cum Rank
        (%)        (%)
       ---- ----  ---- ----
 1.    78.4    1 1297.4    1   CGA_driver_far(934.1s)
 2.    14.0    2  231.6    2   __SysCall_bH(166.7s)
 3.     0.0   54  100.0    3   main(72.0s)
 4.     0.0   19   49.3    5    +Background(35.5s)
 5.     0.0   66   25.3   10    | +SetScenario(18.2s)
 6.     0.0   34   22.8   11    | | +CreateE(16.4s)
 7.     1.4    4   22.8   13    | | | +putimage(16.4s)
 8.     0.0   70    2.5   23    | | +Tunnel(1.8s)
 9.     0.1   15    1.2   31    | | | +line(0.9s)
10.     0.1   16    1.2   32    | | | +setcolor(0.9s)
11.     0.0   20    0.0   35    | | | +bar(0.0s)
12.     0.0   60    0.0   64    | | | +putpixel(0.0s)
13.     0.0   63    0.0   66    | | | +setfillstyle(0.0s)
14.     0.0   35    0.0   48    | | +CreateG(0.0s)
```

would be solved if CritPath had the ability to crawl up through the return addresses on the stack; but the complexity of C stack frames prohibits this. Instead CritPath is obliged to rely on informed guesswork, offering a choice of four strategies which may be applied. These trade off accuracy against CritPath execution time; the manual warns that use of the most precise analysis algorithm may cause the program to run for tens of hours!

Once you know the critical path, you can start work on speeding up your application. This can be done by moving unnecessary code out of the path (the manual gives the example of functions that always return the same value), or by improving algorithms that appear on it. There is also the possibility that the application will turn out not to have any critical path as such. If the reported critical path only takes 10% of execution time, then there is little point in optimising the code in it - as inevitably any improvement will certainly be less than 10%.

CritPath reports its findings in various formats. The most interesting of these is the Weighted Hierarchical Program Flow Tree.

Project managers will find it a useful tool for stamping out individualism among programmers

This is essentially the same as CFlow's flow tree, except that the functions are not ordered alphabetically, but by the amount of time that they consume. The functions on the critical path, therefore, appear at the top of the report. An example of this sort of report is shown in Figure 4; which also illustrates some of the problems with CritPath. The critical path of my program includes a call to a pointer to a function (which confuses CFlow) via a `static` function (which PMon does not know about) into Turbo C's graphics library. CritPath arrives at the conclusion that my program spends 1300% of its time in a non - callable function, `CGA_driver_far()`, which is not quite right. Despite these little hang-ups, CritPath is a powerful and innovative program.

Tool	Purpose
Cat	Superior TYPE command
CFlow	function tree and cross ref
CharCnt	Counts chars, words, lines in file
CLint	Syntax check
CPrint	Pretty printer
CritPath	Program's critical path
CXref	Detailed xref
DeTab	Tabs -> spaces filter
EnTab	Spaces -> tabs filter
ExecTime	Program timer
FileComp	Compare binary files
FileDiff	Compare text files
FileDump	Dump in IBM format
FileList	Superior DIR command
Fill	Batch editor
MapVar	Utility for PMon
PMon	Profiler
Strip	Extract text from file
TabTran	Tabs -> arbitrary string filter
Tail	Copy end of file
TransLit	Any char -> any char filter

Figure 5 - Contents of MS-DOS version of Toolbox

Conclusion

There are a lot of programs in this package; there is not the space to deal with them all. Figure 5 contains a complete list of utilities. I would like to say a little about CLint, which is a lint-ish type utility for picking up stupid syntactic errors in C source code. It queries legal-but-probably-incorrect constructions such as `if (x & y)` (which is usually a typing error version of `if (x && y)`). It isn't as complex as Gimpel Software's PC-Lint - which does virtually everything up to criticising badly-chosen identifier names - but it's not bad. Fill is an off-line editor, blessed with its own programming language. I have not got into this, but it looks very complicated, which proves that it must be good.

A general conclusion on the suite of programs: like MAKE itself, the utilities tend to be something of a pain to set up; but really pay off once you have got them running. The documentation is exemplary; the programs show a consistency of external design which argues the presence of a strong internal design and consequent reliability. I won't call the software 'invaluable', because, let's face it, we have all programmed happily for years without it. But if you were unable to find a good use for at least some of these tools, then my diagnosis would be a lack of imagination on your part, rather than any fault of this package.

.EXE

The C Programmer's Toolbox is available in the UK from System Science (01 833 1022).

Pricing is:
MS-DOS £165
Macintosh MPW £245
Sun UNIX £420 .

These figures exclude VAT but include UK delivery.

PC Magazine Award winner for TECHNICAL EXCELLENCE Development Tools category

INTRODUCING THE REMARKABLE MULTISCOPE DEBUGGER FROM LOGITECH

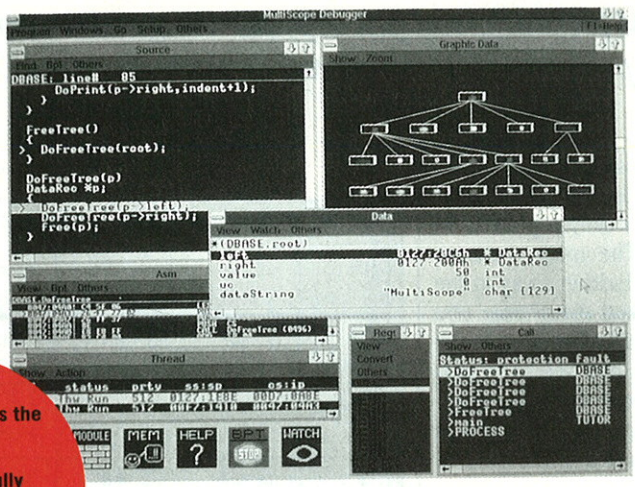
More Debugging Power For OS/2

Logitech's new MultiScope ™ Debugger is quite simply the most powerful and advanced debugging tool available.

This state-of-the-art program, supplied and supported by QA, represents a milestone in debugging technology. Not only will the MultiScope ™ Debugger help relieve the burden of tedious debugging, it is also the only debugger designed to take full advantage of the extra functionality of OS/2.

There's even run-time and post-mortem debugging options, allowing examination of your program both during execution and after a run-time exception.

Never before will you have experienced such a clear insight into your program's inner workings.



Logitech's MultiScope ™ is the **ONLY** debugger to fully utilise the **OS/2** programming environment

USER INTERFACE

MultiScope uses either a Presentation Manager or an OS/2 text mode user interface to provide you with the perfect interface for your requirements. Both execution modes are compatible with the Common User Access of the IBM Systems Application Architecture for maximum ease of learning and use.

DEBUG MULTIPLE LANGUAGES

- IBM C/2
- IBM Pascal/2
- IBM Macro Assembler/2
- Logitech Modula OS/2
- Microsoft C
- Microsoft Pascal
- Microsoft Macro Assembler

2 DEBUGGING OPTIONS

- **Run-time**

MultiScope allows you to monitor the execution of a program with conditional or unconditional breakpoints, and memory or symbolic watchdogs. While you are stepping in your program execution, MultiScope allows you to modify memory or data contents, suspend or resume thread execution.

- **Post-Mortem**

This debugging mode gives all the MultiScope debugging power for examining a program after the occurrence of an OS/2 protection trap, stack overflow, or other run time exception.

GRAPHICAL DATA REPRESENTATION

Only the Logitech MultiScope ™ Debugger permits the graphical representation of your program's data structures. This feature, which operates under Presentation Manager, allows you to explore your dynamically allocated data structures and their internal relationships.

MAXIMUM PERFORMANCE IN TWO MODES

- **Presentation Manager Mode**

MultiScope takes full advantage of Presentation Manager's graphics, multiple fonts, overlapping windows and standard user interface.

- **OS/2 Text Mode**

MultiScope offers a text mode user interface with a very easy-to-use windowing system for the EGA and VGA graphic adaptors. This execution mode is the perfect solution for debugging Presentation Manager applications.

Pull down menus that respond to single-key commands make your debugging faster and easier. Create your own custom configuration of these windows to see exactly what you need. Reduce currently unused windows to icons for easy recall.

MULTIPLE VIEWS

MultiScope gives you 13 different views into your program:–

Source View Look at source code, set breakpoints, find symbols.

Data View Display and modify local and global symbolic data, records, arrays, and set watchpoints.

Graphic View In PM mode, see a graphic representation of any selected data structure.

Assembly View See the assembly decode of the program, complete with the display of corresponding source lines and local and global data.

Thread View Observe threads belonging to your application, and suspend threads, resume threads and update views according to new threads.

Breakpoint View Display the current breakpoint locations, count and limit.

Watch View Display the current watch variables or memory location.

Register View Display and modify the current values of the CPU registers.

Output View Examine the application screen.

Module View Choose between different data object modules of the application and view the corresponding source or data.

Call View Display the source position, local data and parameters of a particular call.

Memory View Display raw memory in various formats; byte, char, text, unsigned word, signed word, address, double word, real, etc.; modify memory contents; set watchpoints.

Log View Log debugger events.

SEND YOUR ORDER TODAY!

ONLY £239

+ VAT including UK delivery.

MORE FEATURES

- **Execution Modes**

MultiScope allows you to execute your program until:–

- the next conditional breakpoint is true
- the next high level instruction
- the next line
- the next assembly instruction
- the return from the current procedure
- the next call from a current procedure
- the next dynamic link load
- the next child process load
- an OS/2 protection violation or other runtime exception
- the next watchdog activation

- **Evaluate Expressions**

MultiScope allows you to evaluate C, Pascal and Modula-2 expressions using symbols defined in your program.

- **Function Calls**

MultiScope is the only OS/2 debugger which can execute calls to your own code during the debugging session. Use your own procedures to print out data structures, or experiment with calls with different parameter values.

- **Child Process Debugging**

When your program executes another program, MultiScope allows you to continue debugging the child process.

- **Dynamic Link Libraries**

The full power of MultiScope is available for debugging your dynamic link libraries.

- **Remote Debugging**

MultiScope allows you to debug a program executing on another computer through a serial line. This feature allows your application to keep all the resources of its computer for its execution.

- **Dual Monitor Debugging**

MultiScope allows you to use a dual mode monochrome monitor for the display of the debugger views while your program is executing on your primary colour monitor.

SYSTEM REQUIREMENTS

- IBM XT/286, AT or 100% compatible
- IBM PS/2 model 50, 60, 70, 80 or 100% compatible
- OS/2 Standard Edition 1.1 or Extended Edition 1.1
- One high density floppy drive and a hard disk

TRY MULTISCOPE FOR YOURSELF

Whilst we've highlighted the main features of this powerful debugging tool, quite honestly the only way to find out about MultiScope is to try it. Call us at QA for a free demo disk containing a complete functioning debugger with tutorial program.

HOW TO ORDER

Call Jan Bevans, Products Group, QA Training Ltd, Cecily Hill Castle, Cirencester, Gloucester GL7 2EF on (0285) 655888 or fax (0285) 650537. Company orders, cheques and Access/Visa welcome. MultiScope costs £274.85 (£239 + VAT) including UK delivery. Call for volume discounts.

QA

QA Training Ltd, Cecily Hill Castle, CIRENCESTER, Gloucester GL7 2EF
Tel: (0285) 655888 Fax: (0285) 650537

CIRCLE NO. 861

A Relocating Loader

To run a .COM program, you just copy the image into memory and jump in. To get a .EXE program going takes rather more effort, as Ken Pillay explains.

Like knicker elastic, a loader is a utility that simply everybody needs, yet is rarely acknowledged and scores only a poor '2' in the glamour ratings. The most common example of a loader is hidden away in the guts of MS-DOS, invoked every time you type in the name of a .EXE program at the 'C>' prompt. Since this one seems to perform its function perfectly well, the question arises: why bother to find out how it works, let alone write a new one?

I originally got involved with loaders while designing an 80186-based embedded system. The machine had to run under its own real time kernel which, unlike MS-DOS, provided facilities such as multi-tasking. The application software was developed on an IBM PC compatible, taking advantage of the large number of tools available for that platform. I needed a program which could load and run .EXE programs on my development system without the help of MS-DOS. This article gives an overview of the theory of loaders, and presents a cut-down version of the program that I wrote to solve my problem.

Two Flavours

There are two types of loader: binary and relocating. They are distinguished by the format of the code that they load into memory, and by their use of the main memory (ie RAM) available. A binary loader, also referred to as an 'absolute loader', is the simpler form. It can load a single program in 'absolute binary form', ie the binary image of the program as it will exist in memory. A program in this form is associated with specific memory locations and hence must be always loaded into the same area of memory if it is to execute correctly.

Clearly, a relocatable program is extremely desirable; in fact John Von Neumann himself was writing relocatable code as early as 1945. Relocatable program files are similar in form to absolute binary files, except that address fields are translated relative to zero,

Figure 1 - Format of .EXE Header

Index (Dec)	File offset (Hex)	Typical values (Hex)	Description
1	00-01	4D 5A	EXE file signature
2	02-03	39 00	Word 0039H = 57, ie the value to be be used to downwards adjust the load module
3	04-05	02 00	0002H = Size of file in 512 byte increments; ie 1024 bytes.
4	06-07	01 00	0001H = number of relocatable items in file.
5	08-09	20 00	0020H = size of header in 16-byte paragraphs; ie header = 512 bytes.
6	0A-0B	07 00	0007H = minimum number of paragraphs required above end of loaded program.
7	0C-0D	FF FF	FFFFH = maximum number of paragraphs required above end of loaded program
8	0E-0F	04 00	0004H = Offset of Stack Segment in load module (in segment form). This value is added to the Start Segment to get the value for SS register.
9	10-11	64 00	0064H = value to be in the SP (Stack Pointer) register when the module is given control. Note 64H = 100, which is the size of the stack.
10	12-13	73 24	2473H = word checksum - negative sum of all words in file, ignoring overflow.
11	14-15	00 00	Value to be in the IP (Instruction Pointer) register when the module is given control.
12	16-17	00 00	Offset of Code Segment within load module.
13	18-19	1E 00	001EH = Offset of first relocation item in file.
14	1A-1B	00 00	Overlay number = 0. Therefore program is resident.
15	1C-1D	01 00	Undocumented/reserved.
16	1E-1F	05 00	0005H is the Offset value that points to Relocation item 1.
17	20-21	00 00	0000H is the Segment value that points to Relocation item 1.

PANEL® *Plus* II

Screen Generator and User Interface Library for 'C'

PANEL *Plus* II features: Interactive screen design editor • C code generator • extensive library of user-interface functions • pop-up fields/windows • multiple-line fields • horizontal/vertical scrolling regions • scroll bar/mouse support • help boxes • custom validation • pull-down menus • complete **library source** included • support for popular **graphics libraries** • OS/2 DLL • **no royalties.**

Just released: **Utility Source Option** including source code for the screen layout editor and code generators, which allows **PANEL *Plus* II** to be ported to any suitable development platform. This gives you the freedom to choose and switch your development environment, with total screen and source code compatibilty . Also now supported: Microsoft **FORTRAN** under DOS and OS/2 (the DOS version requires the MS-C run-time library).

Panel *Plus* II can now run on any system with 'C' (but block-mode terminals are not supported). Cost per user: **£295** (including 640-page manual). Please call for quantity pricing, full version list, and free demo disk.

Roundhill Computer Systems Limited

Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU
Telephone: **0672 84 535** • Fax: 0672 84 525 • BIX: *join roundhill*

CIRCLE NO. 862

and relocation data is stored with the program.

There are also two general approaches to encoding relocation data. In one approach, the language translator (ie the compiler or assembler) appends a relocation bit to each machine language instruction. This bit is set by the translator only if the address fields of the corresponding instruction are to be relocated. This method is obviously very hardware-dependent, and can only be used with machines that implement direct addressing.

In the second approach, all relocation data is grouped into a 'relocation table' (or 'dictionary') by the linker component of the compiler. The relocation table contains a pointer to each machine language instruction that must have its address field adjusted. This latter approach is used in systems that use base addressing or segmentation, and is the method used in MS-DOS .EXE files.

It is perhaps worth saying a word or two about the .COM format. You will recall that two types of binary files (.EXE and .COM) are executable under the MS-DOS environment. The .COM format is a hangover from CP/M, when the OS loaded *all* executables at absolute address 0100H (CP/M machines had only 64 KB of address space, arranged linearly from 0000H to 0FFFFH. The OS lived in high memory, while the first 100H bytes were reserved for such things as the operating system call address).

The MS-DOS version of the .COM file is also an absolute memory image of the program to be executed. However, it *does* offer a degree of 'reloctability', since (in theory) .COM programs don't 'know' about segments. All the segment registers are fixed up to the same value before execution begins. Such programs are a maximum of 64 KB in size and conform to the 'small memory model' of the 80x88/6. A .EXE program can use the 'medium' and 'large memory model' of the 80x86 where the program has different segments for code, data and stack (NB: this use of the small/medium/large model terminology is different from the way it is used by C compiler vendors). There can also be multiple code and data segments.

File Structure

A .EXE file consists of two parts: a header and the load module. (This article deals with only the MS-DOS version of the .EXE file structure; the OS/2 superset is not covered.) The control and relocation information is contained in the header. This consists of a formatted structure, from offset zero to 1BH, a variable sized space reserved by MS-DOS and a relocation table. Figure 1 shows the structure of the formatted part of the header, with example contents. Following this is the variable-sized space reserved by MS-DOS. Ray Duncan's book *Advanced MS-DOS* states that it is a reserved variable space. Examination of various .EXE files shows that it consists of one word at offset 1CH-1DH, which seems always to contain the value 0001H.

Next comes the relocation table itself (its actual position and its length given in the formatted header). The table is made up of a variable number of 'relocation items', each containing two fields: a 2-byte offset value followed by a 2-byte segment value. These point to the offset into the load module of a word which requires modification before the load module is given control.

How it Works

Figure 2 shows a simple MS-DOS assembly language program to load a .EXE file. It can easily be adapted to run on non MS-DOS hardware. For example, the program reads in one byte at a time in sequence from the file, so the conversion necessary to make it load from, say, a serial port is trivial. Needless to say, it does not cope with overlays. Here are the steps performed by the relocating loader.

Since MS-DOS's loader works perfectly well, the question arises: why bother to write a new one?

- The program opens the .EXE file, hard coded as the variable `fname` in Figure 2.
- A check is made for the .EXE file signature, ie the word 5A4DH at the start of the file. The program aborts if it is not found.
- The rest of the formatted header is read from the file and stored.
- The program calculates the size of the header in bytes.

Figure 2 - The .EXE Loader Program

```
;PCLOAD
;Loads and runs a .EXE file,
;without using EXEC function call


cr equ 0DH
lf equ 0AH

;Image load absolute address

Startseg equ 3000h

;return errors

notEXE equ 1
IOerr equ 2

dseg  segment

;**** Test filename
fname   db 'GWBASIC.EXE',0



;**** Formatted EXE Header

EXEsig          dw 5A4DH
headerstart:
ldsizeDAF       dw 0
filesize        dw 0
nreloc          dw 0
headsize        dw 0
minpara         dw 0
maxpara         dw 0
SSadd           dw 0
SPoint          dw 0
Chksum          dw 0
IPoint          dw 0
CSadd           dw 0
Reloc1          dw 0
Ovlno           dw 0

reloctab dw 500h dup (0) ;Relocation items

count    dw 0
Loadsize dw 0
IPCS     dw 2 dup(0)

databuf    db 100 dup(0)
filehandle dw 0


Uldsize  dw 0
Lldsize  dw 0
Ucount   dw 0
Lcount   dw 0

dseg  ends

sseg  segment stack
     db 100 dup (0)
sseg    ends

;*** PROGRAM CODE

cseg  segment

assume ds:dseg,cs:cseg,ss:sseg

;Entry point
xload   proc  far
  mov   ax,dseg
  mov   ds,ax
  call  open    ;open EXE file


  call  readword  ;get EXE signature
  cmp   ax,EXEsig
  jz    continue
  mov   al,notEXE  ;Not EXE file error
  jmp   endprog
```

```
continue:

;Store header from offset 02 to 1B of EXE file
  mov   cx,(1bh-02h)/2+1
  push  ds
  pop   es
  mov   di,offset headerstart
  cld
gtheader:
  call  readword
  stosw
  loop  gtheader


;**** Get Header size
;(Assume less than 64 KB)
;
  mov   ax,010H
  mul   headsize
  mov   headsize,ax
;headsize now in bytes


;**** Get Load Module size


  mov   ax,0200H  ;= 512 decimal.
  mul   filesize
  mov   Uldsize,dx
  sub   ax,headsize
  sbb   dx,0
;the loadsize must now be adjusted
;downwards by variable ldsizeDAF
  sub   ax,0200H
  sbb   dx,0
  add   ax,ldsizeDAF
  jnc   addDAF
  inc   Uldsize


addDAF:
  mov   Lldsize,ax
;


;**** Read in relocating items into a table


reloc:
  mov   ax,Reloc1
  cmp   ax,count  ;count contains no of bytes
          ;read from file
  jz   rdreloc
  call  readbyte  ;dummy read
  jmp   reloc


rdreloc:
  mov   cx,nreloc
  mov   bx,0h
rdreloc1:
  call  readword
  mov   reloctab+[bx],ax ;get offset
  add   bx,2
  call  readword
  mov   reloctab+[bx],ax ;get segment
  add   bx,2
  dec   cx
  cmp   cx,0
  jnz   rdreloc1


;**** find start of load image in file


findload:
  mov   ax,headsize
  cmp   ax,count
  jz   loadimage
  call  readbyte  ;dummy read
  jmp   findload


;************** read in load image
loadimage:
  mov   bx,0    ;load image at
  mov   ax,Startseg ;address Startseg:0
  mov   es,ax
  mov   ax,Lldsize
  mov   Lcount,ax
  mov   ax,Uldsize
  mov   Ucount,ax
;
loadimage1:
  call  readbyte  ;call preserves bx
  mov   es:[bx],al
  add   bx,1
  jnc   ld1
  mov   ax,es
  add   ax,1000h  ;bump ES to next seg
  mov   es,ax


ld1:
  sub   Lcount,1
  jnz   loadimage1
  cmp   Ucount, 0
  je   loadexit
  dec   Ucount
  jmp   loadimage1


loadexit:


;**** Apply fix-ups to relocating items


  mov   bx,0h
  mov   cx,nreloc
fixup:
  mov   si,reloctab+[bx]
  add   bx,2
  mov   ax,reloctab+[bx]
  add   ax,Startseg
  mov   es,ax
  mov   ax, WORD PTR es:[si]
  add   ax,Startseg
  mov   WORD PTR es:[si],ax
  add   bx,2
  loop  fixup


;**** Set up stack


  mov   ax,SSadd
  add   ax,Startseg
  mov   SS,ax
  mov   SP,Spoint
  mov   ax,CSadd
  add   ax,Startseg
  mov   CSadd,ax


;**** Jump to entry point of program


  jmp   dword ptr Ipoint


;**** Program in EXE file executed !!!


  mov  al,0     ;return ok


endprog:
  mov  ah,4ch    ;MS-DOS exit fn
  int  21h
xload endp



;**** Supporting Subroutines


readword  proc    ;returns word in AX
  push  cx
  call  readbyte
  push  ax
  call  readbyte
  mov   ah,al
  pop   cx
  mov   al,cl
  pop   cx
  ret
readword  endp


readbyte  proc    ;returns byte in AL
  push  bx
  push  cx
  mov   ah,03fh
  mov   bx,filehandle
  mov   cx,1    ;no of bytes to read
  mov   dx,seg databuf
  mov   ds,dx
  mov   dx,offset databuf
  int   21h
  mov   al,databuf ;copy byte to AL
  jc   errexit
  pop   cx
  pop   bx
  inc   count
  ret
readbyte  endp


open    proc    ;opens file
  mov   ax,3d02h
  mov   dx,offset fname
  int   21h
  jc   errexit
  mov   filehandle,ax
  ret
open    endp


errexit  proc     ;I/O error exit
  mov  al,IOerr
  jmp  endprog
errexit  endp


cseg    ends


end    xload
```

- The program calculates the exact load module size in bytes. This is done by taking the `filesize` in bytes, subtracting the header size from it, and downwards adjusting it by the 'Downwards Adjusting Factor' given in the header.

- The program reads from the .EXE file until the first Relocation Item is reached. All the items are then read into a relocation table.

- The program reads the load image into memory, starting at a location defined by the equate `Startseg`. If you are testing the program on a PC, then I have found that a value of 3000H seems to work, provided you haven't overdone the TSRs.

- Relocation is performed for all relocation items. Each item's segment value has `Startseg` added to it. The item now points to a word in the load module. The start segment value, `Startseg`, is added to this word and the result is placed back in the word in the load module.

- The Stack Segment register, SS, and the Stack Pointer, SP, are set using values in the formatted header, once again adding `Startseg` to the segment value. The program entry point is calculated in the same way.

- A far jump to the entry point of the loaded program is performed.

Conclusion

Software development for 80x86-based real time systems can be accomplished using the 'target' development system, ICEs and various expensive hardware and software available from Intel and other companies. Using your own loader, you can develop the application on a nice, cheap PC running nice, cheap MS-DOS. You can write in Turbo Pascal, Turbo C, JPI Modula-2 or whatever takes your fancy. Then, when you have got them going nicely, you port the whole .EXE file down to the target and load it *in situ*. Neat, eh?

.EXE

Ken Pillay is a member of the Faculty of Engineering at the University of Exeter, where he teaches Electronics and Computing. His research interests are in microprocessor systems, LANs and semi-custom design. He can be contacted at the Faculty of Engineering, University of Exeter, North Park Road, Exeter EX4 4DF or on the JANET mail system as PILLAY.KDA@UK.AC.EXETER.

dBASE LANGUAGE FOR VAX, UNIX & XENIX

RECITAL, a comprehensive relational database management system and fourth generation development environment, is TOTALLY upwards compatible with Ashton Tate's dBase III PLUS. Now you can migrate your PC applications to your VAX, or a wide variety of UNIX and XENIX platforms—*without reprogramming!*

RECITAL lets you upload dBase applications and associated data DIRECTLY from your PC to your VAX or UNIX machine. And once it's installed you can maintain and modify it using the same FAMILIAR syntax and commands as on the PC. Operators trained in dBase will understand RECITAL *instantly.*

The potential is exciting. Not only is the whole catalogue of affordable dBase applications yours to use, but you can easily develop entirely new programs using RECITAL. And since there are thousands of programmers already trained in dBase, finding skilled people becomes much easier. Not only that, with RECITAL, end users can develop and maintain databases, generate reports, design screen forms, and much more . . . without writing *any program code at all.*

RECITAL is functionally rich

- ☐ Upwards compatible with
  - dBase III Plus,
  - FoxBASE Plus
  - Clipper
- ☐ Multi-user relational DBMS
- ☐ Concurrent processing
- ☐ Transparent access to existing RMS files
- ☐ Powerful fourth-generation language
- ☐ Complete menu driven end user interface
- ☐ Powerful forms based tools
- ☐ Comprehensive report writing language
  - screens
  - reports
  - labels
- ☐ 2-way interface to 20/20 spreadsheet
- ☐ Keyboard macros usable in any form or menus
- ☐ Flexible prototyping tool
- ☐ Active data dictionary with full referential integrity constraints

RECITAL *complements* other software products and helps you *release* their full power.

For further information please contact:
RECITAL CORPORATION LTD
South Bank Technopark
90 London Road
London SE1 6LN
Tel: 01-401 2727
Fax: 01-633 9617

PROVE IT FOR YOURSELF!
Contact Recital Corporation today for details of our 30 day evaluation.

ANSI C-Compilers

True processor specific, and not just modified generic, C-compilers are now available.

1802, 1805, 6301, 64180, 6502, 6800, 6801, 6802, 6805, 6809, 6811, 68000, 8051, 8085, 8086, 8096, Z80.....

Matching Assemblers, Linkers and Development Systems with High Level Debugging are also available to see you through your complete software and hardware design cycle.

Call us now for your **FREE** active demonstration disk.

american automation

uk oxford

Tel: (0993) 778991 Fax: (0993) 702284

Lea View House, Two Rivers Est, Station Lane, Witney, Oxon OX8 6BH

CIRCLE NO. 868

USING INFORMIX?

The E.I.U.G. is an independent non-profit making group of Informix users which acts as a forum for the exchange of information, ideas and experiences. We publish an informative newsletter and hold regular meetings for members. Our next meeting is on the **17th May 1990** in central London.

For further details contact

Bryce Campbell on 0532 738244
or Pat Lloyd on 0628 822266

EUROPEAN INFORMIX USERS GROUP

Informix is a trademark of Informix Software Inc.

CIRCLE NO. 869

THE MICROSOFT PROGRAMMERS LIBRARY

Comprehensive CD-Rom database of 72 Microsoft technical manuals and best selling reference books that can be popped up in your favourite editor.
3000+ sample programs broken down into thousands of files.
Ideal for those learning as well as professionals.

THE BENEFITS

Reduce valuable programming time/meet deadlines.
Sample codes are indexed and referenced and can be copied or pasted directly into your editor.
Customised bookmarks allow you to define upto 80 'often used' reference points.
Development teams can be networked to one CD-ROM drive.
Use 'hot key' to 'pop up' for quick referencing.

THE REFERENCE CATEGORIES

- Windows
- OS/2
- Macro Assembler(MASM)
- 'C'
- MS-DOS
- Pascal
- FORTRAN
- BASIC
- Hardware

THE SUPPORTED EDITORS

- BRIEF 2.0
- Epsilon 3.21
- QuickBasic 4.0
- Microsoft
- Norton 1.3c
- QuickC Editor 1.0
- Turbo Pascal
- TurboC
- VEDIT PLUS

LIBRARY CD £295 *CD DRIVE £595* *LIBRARY + DRIVE £799*

All prices Exclude VAT

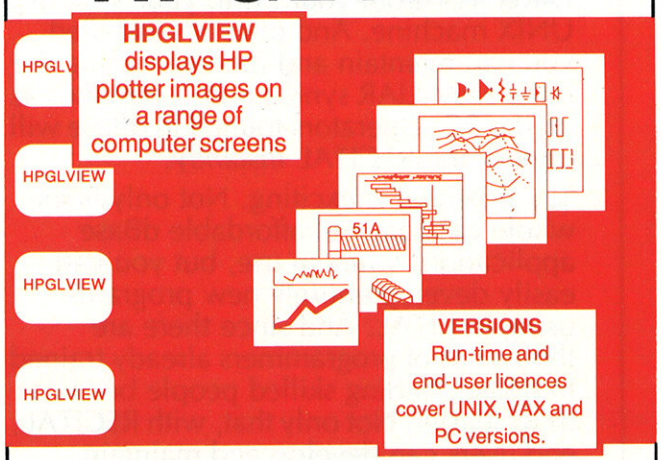
SOFTLoK International Limited

Units 10 & 11, Railway Road Industrial Estate, Darwen, Lancashire, BB3 3EH, England.
Tel (0254) 772220 Fax (0254) 705956

VISA

CIRCLE NO. 870

HPGLVIEW



A (run-time) tool for software developers

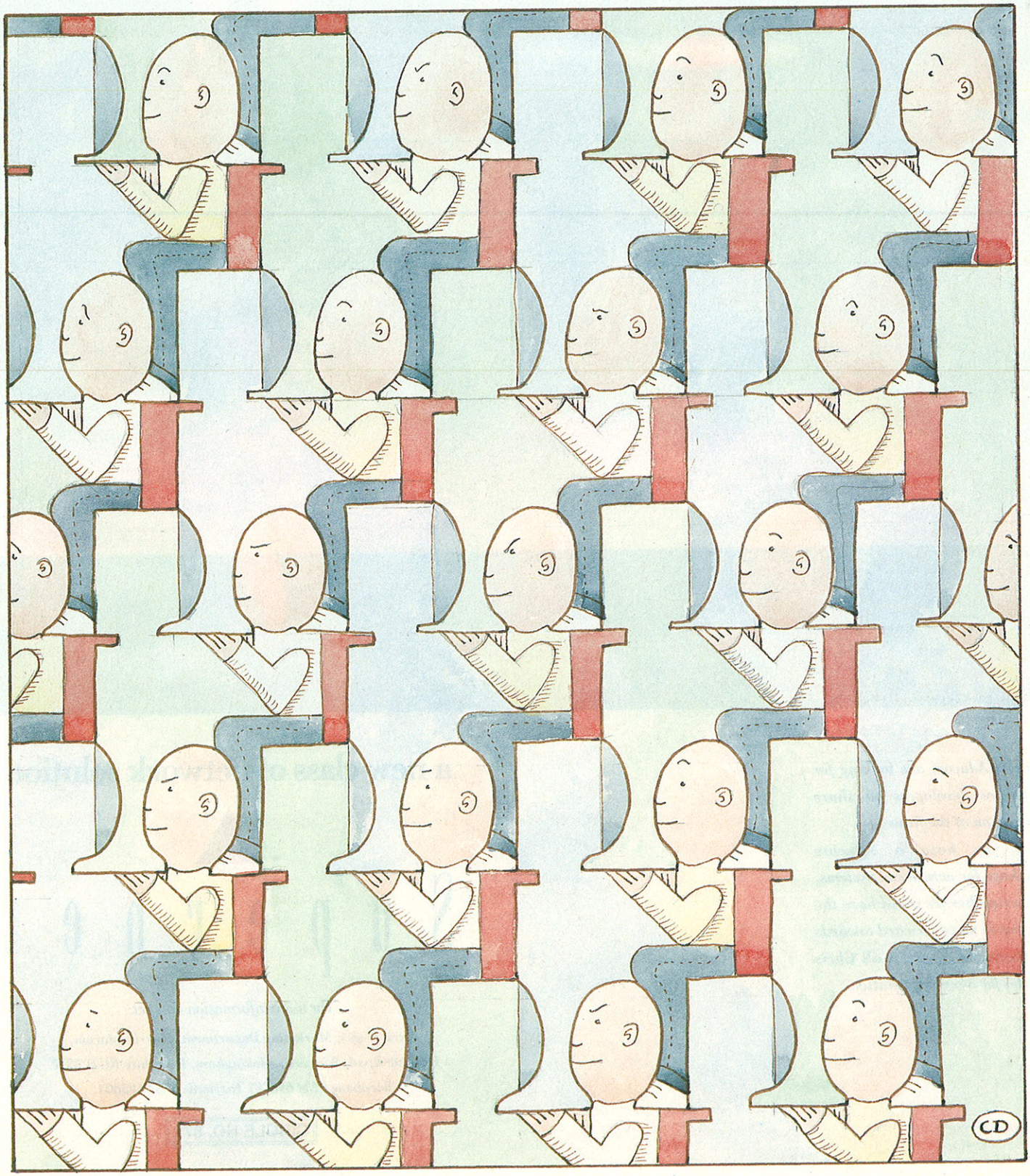
Take advantage of the general availability of HPGL for simple graphics display and distribution. **Plotting applications ported from other sytems or otherwise without screen graphics will come to life when their plot files are displayed by HPGLVIEW.**
For impatient plotter users and software developers who want to check plots without requiring hardcopy, end-user versions are available - for the IBM PC/AT HPGLVIEW is ex-stock at £100 per user.
Call now to discuss how HPGLVIEW can reduce the time spent on unnecessary plotting by you and your customers.

CAMEL SERVICES Ltd • OXFORD (0865) 512678

CIRCLE NO. 871

SPECIAL FEATURE

NETWORKING



For the business of today with an eye for the future

Harris Adacom are looking for software developers to share our vision of the future.

If you have a software package for networked systems, then together we could have the means to move forward towards a successful future. Call Chris Wyles for more information.

a new class of network solution

Supernet

For more information contact:

Chris Wyles, Marketing Department, Harris Adacom, Eskdale Road, Winnersh, Wokingham, Berkshire RG11 5TR.

Telephone 0734 698787. Facsimile 0734 695805.

CIRCLE NO. 872

Networks for Software Development

What are the benefits of networking a software development team? Andy Redfern asked two software houses to explain what, if anything, they find the most useful.

Unquestionably, the last decade was the era of personal computing. The pundits are now predicting what the 90s have in store for us all. Steve Jobs, founder of NexT Inc, recently predicted that it will be the decade of 'interpersonal computing'. Software will be replaced by groupware. Stand alone PCs will be networked. The whole office environment will be geared towards information and communication. But what effect will this have on the programmers who develop the software?

Through the media and by word of mouth, the legendary programmer is portrayed huddled over his computer keyboard. He works alone and at night. Of course, like all myths there is an element of truth in such a picture, but today's programmer is often a professional who works as part of a development team. 'Interpersonal computing' or networking would seem to offer benefits for both the development team and the users of the end product. But do the benefits outweigh the disadvantages of cost and loss of individuality? Are there enough network products to provide software developers with the specialist tools that they need?

In search of the answers to these questions, I talked to two very different, but equally successful UK software houses - Intuitive Systems in Stevenage, and Zortech in Woolwich.

Intuitive Systems

Over the last five years, the team of six programmers at Intuitive Systems have developed a range of database products. Their products are aimed at the corporate PC network environment and, therefore, they needed to install a network to ensure that their final product actually worked. The Company have since discovered that the network also plays an important role in the efficiency of their development work itself.

All the PCs used by the team are networked using 3Com Etherlink cards. The network unusually supports two servers - one running Novell Netware and the other running OS/2 LAN Manager - over the same cabling. As a machine boots onto the network, the user can select which server to use depending on which operating system they are using.

All of the development work is done in the Glockenspiel C++ translator using the Microsoft C compiler. All C++ and C header files, compiler support files and libraries are stored on the remote server. This allows a single backup of all the development work to be taken regularly, using a single tape streamer. Before the network was installed, one tape streamer was shared by the whole team. Naturally, this involved a considerable amount of time being wasted installing and un-installing the streamer, just to make backups of each person's hard disk.

Problems

The biggest problem in having only one copy of all the files on the server is how to control the use and modification of the files by six different programmers. Andy Bowyer, Project Leader at Intuitive Systems, chose Polytron's version control software PolyPVCS and their network MAKE utility, PolyMake. 'Initially, we all worked completely off the networked drives - headers, source, libraries and .EXE files were all called from the remote drives. The only problem with that was that the MAKE process slowed dramatically as the network traffic increased. We now use PolyPVCS to book out source and header files, and keep the copies on our local hard disks. This speeds up the MAKE process quite dramatically.'

The documentation for their products is also produced on the network using Microsoft Word for OS/2. This allows the programmer who created the relevant section of the product to document it directly in a single copy of the manual. The network is also used to give everyone access to the printing resources, but the other network buzzword, email, has failed to impress any of the development team.

Running the two networks side by side has given Intuitive Systems an interesting insight into how the two products compare with each other. LAN Manager's biggest problem seems to be speed - it provides an average access time of little more than twice the speed of a floppy disk. But the advantages of LAN Manager for a software developer are significant. Writing a server application under Netware is possible, but all the routines, protocols and techniques used are proprietary, making porting the software to another network difficult. Using LAN Manager it is a relatively uncomplicated process. In fact, it is similar enough in operation to UNIX that transferring servers between the two environments is an elementary task. Also the server's standard interface means that users of the Intuitive Solution product can not only gain access to their data using Intuitive's own server, but also through IBM and Sybase's SQL servers. A distinct advantage for gaining access to the IBM-dominated corporate arena.

Everything sounds too good to be true? Andy Bowyer admits there are disadvantages. 'If the server crashes, or fails to boot

in the morning, no one can do any work until it's sorted out. Crashes are less frequent with LAN Manager than with Netware, but they still occur and it affects everyone when they do'. Obviously, this problem has been lessened to some extent by the introduction of PolyPVCS for booking out files to local drives.

Zortech

The very fact that Zortech has grown as a company pays tribute to the dramatic advances in communication and network technology. Zortech's head office may well be in Woolwich, but technical support is in Yorkshire, one of the support programmers operates from Grantham, most of the documentation is written in Hertfordshire, the US office is based in Boston, while Walter Bright, the creator of Zortech C and C++, works at the opposite end of the States in Seattle. Without an excellent communication structure the company would be strangled by delays, and products not being in the right place at the right time.

All the PCs in the Woolwich and Boston offices are networked together using a Novell system. These networks run all their accounts and word processing, and are used in the production of the compiler documentation. On both sides of the Atlantic, Zortech use Sage Multi-user Accounts for its simplicity and the excellent reporting facilities it provides.

It all sounds very sensible and fairly mundane until John Haggins, Managing Director of Zortech, begins to wax lyrical about the email system they use. 'We use an electronic mail system called Coordinator 2 from Action Technologies in the States. It provides a message service for everyone on the two networks, and for all those who work from home'.

Basically, Coordinator 2 can either run on a networked system or on a stand alone machine, transmitting the messages across the network links or through a modem as needed. For example, if someone in Head Office wants to send a message to Walter in Seattle, they simply enter the message and address it to him. The UK system will then automatically dial the Boston office and transfer the message across. At a later time, Walter would dial into the Boston server using the stand alone version of Coordinator, and download all his messages. Then, while off line, he can prepare replies to the messages and log on once more to the Boston system to send them.

John Haggins continues, 'It's amazing, I don't have to repeat myself any more. If I have an interesting phone call that the US Managing Director and others should hear about, I simply mail him and send copies to everyone else. The system even allows us to attach files to the messages'.

In the Summer and Autumn of 1989, Zortech set themselves a difficult challenge - to turn the specification of C++ version 2 from a paper document in June, to a finished shrink wrapped product in November. Haggins claims that one of the main reasons that was possible was through using the email system. 'Even though we were 8000 miles apart, it was like Walter was in the office next door. The only reason that the November shipment date was met, was because of the communications technology we employed.'

More Problems

Of course, the system isn't perfect. If a message and six copies are sent to people in the States, then the same message is transmitted across the Atlantic seven times. If a file is attached, that gets sent seven times as well. This is symptomatic of the main problems facing network users - the utility and specialist software that is so abundant on the stand alone PC just isn't available on network systems yet. Special tasks have to be written by yourself, or not done at all.

For example, Sage Multi-user Accounts, like most network software, employs file locking because it's simpler than the more efficient record locking. This means that a network user will regularly be denied access to a file because someone else is using it. Worse than that, it's often impossible to find out exactly who is using it. So frustrated were the staff on the network that, in the end, one of the team wrote a little pop-up utility called 'Who's got it?'.

In another instance, a utility was written to send a copy of the day's trading to the States. John Haggins maintains that it is a worthwhile product to use, and it certainly saves him time, but it just isn't cost effective to write that kind of software yourself. 'There's a tremendous opportunity out there for someone to write some excellent network products. The problem is either that people are not doing it, or the dealer/distributor network is not taking on such products.'

Conclusion

On balance, it would appear that networks do provide benefits for software developers and their support teams who market the products they write. Networked version control software and MAKE utilities are obviously a real bonus and, in the distributed company, a powerful email tool will soon become indispensable. But the problems are not insignificant.

The network takes away some of the programmer's independence, with network and server crashes strangling the development process until they are fixed. But even these problems are bearable compared with the dramatic lack of software products that not just 'support' networks, but actually make efficient use of them.

The problem is exacerbated because software development is a minority business activity compared with database access, spreadsheet usage and word processing activities. There is evidence that software distributors and dealers are taking networks more seriously, but the problem will only be solved when the number of installed networks is large enough to make it worthwhile for programmers and developers to produce network products. .EXE

Andy Redfern worked as a staff writer for PCW Magazine until late last year. He is now a freelance computer journalist.

PRODUCTS MENTIONED

Novell Netware
Price: £400-£4000 depending on system
Tel: 0344 860400

Etherlink Cards
3Com
Tel: 0628 890670

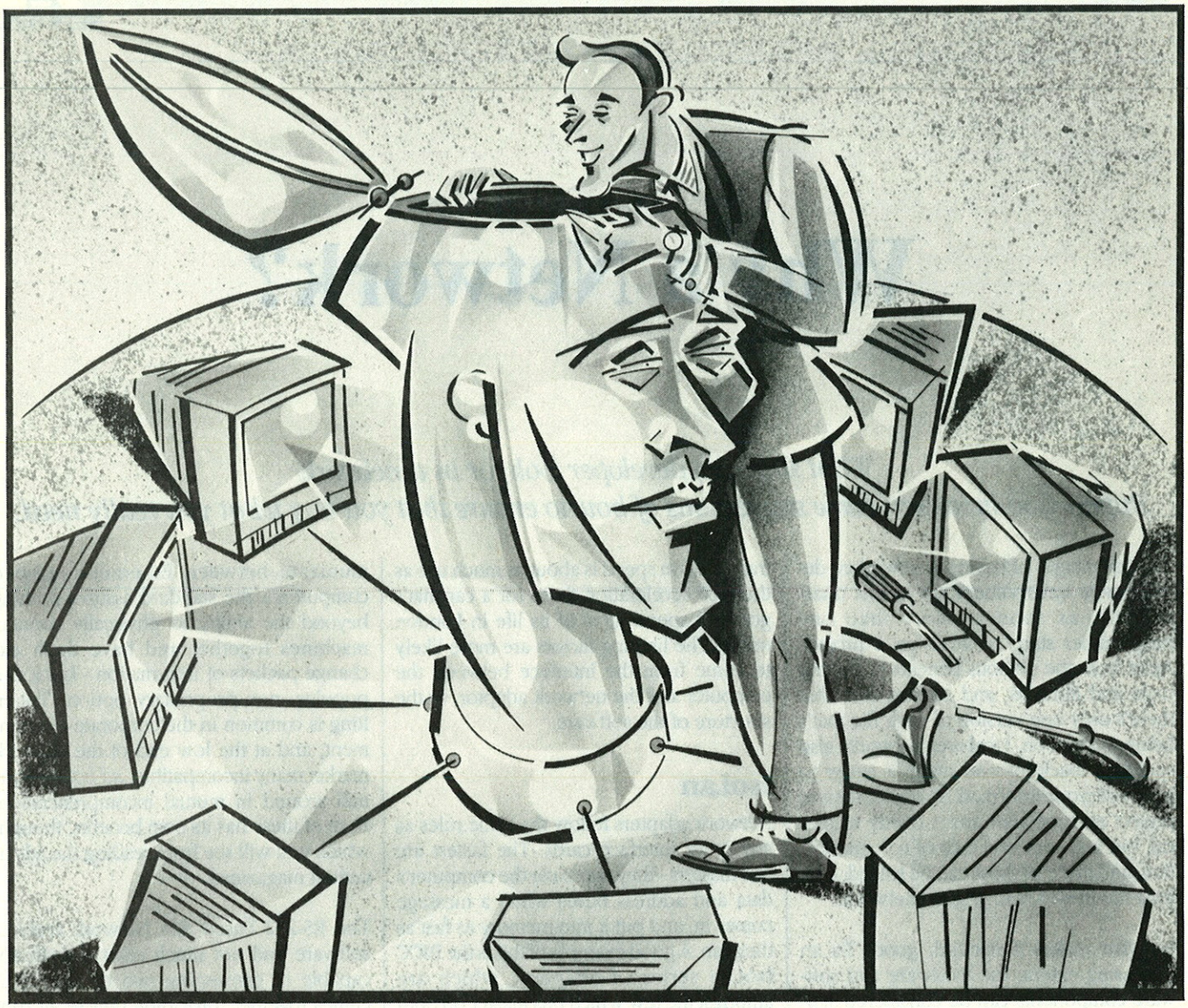
Coordinator 2
Action Technologies
Tel: 0101 415 654 4444

Sage Multi-user Accounts
Sage PLC
Tel: 091 213 1555
Price: starts at £349 + VAT

OS/2 LAN Manager
Various OEMs
Apricot
Tel: 021 456 1234
Price: £895 1-5 users, £1495 1-10 users and £3250 for the full version with no limit on the number of users

Word for OS/2
Microsoft
Tel: 0734 391123
Price: £395 + VAT

PolyMake and PolyPVCS
Software Generation
Tel: 0462 422525
Price: 5 User PolyPVCS £1449 + VAT
5 User PolyMake £409 + VAT



See us at
WHICH? COMPUTER
Stand No. N131

IPX or Netbios networks examined through remote controlled PCs

NetOp from Richmond Systems offers fast and easy troubleshooting for the network manager.

Thanks to a new software concept, it is now possible to monitor and control multiple PCs on a local area network either from a single workstation on that network or remotely using modem links.

NetOp combines the following valuable features:

- *Uses only 1 to 2 kbyte of memory on each remote station*
- *Full graphics support including EGA and VGA*
- *Multiwindowing management facility*
- *Password and keyboard lock facilities*
- *Operates in conjunction with established remote communications packages*

This simple-to-install package has an international reputation and has already proved its value throughout Europe and Scandinavia.

To find out more about NetOp and details of your local dealer please contact:

Richmond Systems Ltd
3, Church Terrace,
Richmond, Surrey
TW10 6SE

Tel: 081-940 6986
Fax: 081-948 2026

NetOp

CIRCLE NO. 873

Why a Network?

What should a developer look for in a network?
Rupert Goodwins has some suggestions of how to ensure that you buy what you really need.

The advantages of having a networked development environment are almost unassailable, as soon as more than one programmer starts to work on a project. Apart from the obvious benefits in sharing tools and libraries, and making sure that there's only one version of each file that's being worked on, local area networks also speed up backups and make it easier to keep everyone informed. Most problems in project management aren't overly technical, but arise out of a lack of communication, and many of these can be solved, or at least minimised, with a good network.

So what makes a network good? To an increasing extent, the hardware and software components can be considered separately, since the same process of standardization that personal computers went through is well under way. I'll start with hardware.

Ethernet

An off-the-shelf Ethernet card will come with various drivers which interface with off-the-shelf network software packages. Make sure that any card you buy offers compatibility with Novell Netware, 3Com and the NetBIOS. If you're intending to use the system with a popular application, such as PVCS version control software, you'll probably find that almost anything will work. Of course, some will work faster than others.

One of the figures that often gets bandied about when men in suits try to sell networks is the speed in bits per second. Cheap networks start at RS-232 speeds, using RS-232 wiring (The Invisible Network, and The $25 Network, for example), while the latest standards talk about fibre optics. Far from being the most important factor in the rapidity of the network, though, the raw transmission speed is about as much use as the 0-60 acceleration times for a car that's going to spend most of its life in London traffic. The limiting factors are more likely to come from the interface between the computer and the network adaptor, or the structure of the software.

IsoLan

Network adapters follow the same rules as any other interface cards. The fastest are bus masters - they take over the computer's data and address buses when a message comes in, and put it into memory as fast as they can. A good example of this is the BICC IsoLan series of adapters, which are amongst the few which bus master on both MCA - which is designed explicitly for this - and AT-style slots.

The next fastest category of cards are those that use programmed I/O; this is the most common sort of card, and relies on the host computer to issue explicit I/O commands to read in the data. Last, and definitely least, are those that rely on DMA. Direct Memory Access is similar to bus mastering, in that the network card can place information into the computer's memory directly, but it relies on circuitry within the PC which suffers from being slow, complex to use and has been known to harbour incompatibilities. Only on XT machines is it faster than programmed I/O, and then not by much.

Standards

The different network topologies and hardware standards are still miles apart. The small systems standard is Ethernet cum IEEE 802.3, which defines electrical and other interface specifications used by most PCs and almost all UNIX machines. You'll still need the appropriate software to communicate between Ethernet-equipped computers - the standard assures nothing beyond the ability to physically connect machines together and have them exchange packets of information - but it is a popular, non-proprietary option. Token Ring is common in the corporate environment, and at the low end of the network market many incompatible, *ad hoc* systems mill around in mutual incomprehension. Each of these has its own benefits, though, which you will see from reading the ads in certain magazines.

The RS-232 based $25 Network (cables, software and not much else) is perfectly capable of connecting two or more machines together, providing they're all running exactly the same software, and some other, much faster, cheap and cheerful networks have the same limitation. The rule of thumb is that if you're paying £200 or less per node for hardware and software, you're losing compatibility.

Software

Network software is another, and more complex matter. To the user, the developer and his applications, any networked devices should seem identical to those physically present on his machine. In practice, network software takes up some memory, and if you're short of RAM on a pure MS-DOS machine this can be significant. The amount of RAM lost ranges from about 15 KB for just MS-DOS file functions, to over 100 KB if you want full printer support, disk and printer sharing, and electronic mail. Also, memory usage varies from product to product, and an increasing number make good use of extended memory to leave the MS-DOS space as clear as possible. Some, like Lantastic, have RAM on the adaptor board itself which houses some of the network software.

WE MAKE THE PIECES FIT!

pcANYWHERE™ III

Remote Computing at its best:– £120

The easiest way to resolve user problems from afar: Corporate Information Centres, Software Houses and Computer Dealers can set up diagnostic, support and operational systems, including access to Local Area Networks via dial-up and leased lines. The pcANYWHERE III package includes software for both the host and remote sites.

★ Full control of a host pc from a local pc or terminal allows almost any pc software to be used from a remote site.

★ UK version, with support for wide range of available modems (eg Hayes, Amstrad, Dowty, Miracom, Pace, Tricom and Dacom) including V21, V22, V22bis, V32 and V42 standards.

★ Graphics support for CGA, EGA, MCGA, Hercules and VGA.

★ Help service direct from DMA's authorised UK distributor, including special modem configuration service.

★ Call-back and password security.

★ Fast error checked file transfer.

Call now 02406 6365

UK VERSIONS

Even use an Apple Macintosh running **pcMACTERM (£99)** to control a host pc running pcANYWHERE III, transfer files, cut and paste pc screens into Mac applications, print on either pc or MAC printer.

And for remote control across a local area network, **NetOp (£345)** for NETBIOS compatible networks, and **pcMACTERM/Network (£345)** for Appletalk networks.

CHAIRMAN™

Set up your own Electronic Mail, Bulletin Board or Information System quickly and painlessly using

CHAIRMAN... £210 TO £710

Supports up to 6 simultaneous lines on one pc, or unlimited callers in network version. Security checking, File transfer, Conferencing, On-line Surveys, Binary Mail transfer, Fully customisable even to the extent of running Database applications.

ASCOM IV™

ASCOM IV

For General purpose communications via modems or direct connection, this powerful menu or command-driven package allows easy use of Electronic Mail systems, access to mini's and mainframes and other on-line services. Dial directory, Powerful script language, selection of file transfer protocols including XMODEM, YMODEM and Kermit. Terminal emulation includes VT220, VT100 and Data General D200, and can even be used as a terminal to pcANYWHERE III . . . **£170**

TO MAKE YOUR CONNECTIONS RELIABLE AND FASTER HAYES SMARTMODEMS ARE THE ANSWER

V-Series 9600 + Quad	**£999**	**2400 V22, V22BIS**	**£499**
V-Series 2400 + Quad	**£649**	**1200 V21, V22, V23**	**£399**
2400 Quad (V21, V22, V23, V22BIS)	**£599**	**1200B (Internal)**	**£349**

Trade and OEM enquiries are welcome.
Standard terms are CWO. Access and VISA cards accepted.
Prices exclude delivery and VAT.

Hayes™ DMA Authorised distributors

Telesystems Ltd

3 Wycombe Road, Prestwood, Bucks HP16 0ND. (02406) 6365

CIRCLE NO. 874

There's a big difference in the amount of memory required for redirectors and that for servers. Redirectors - increasingly called workstations, just to confuse things - are computers that run application software and rely on the network to provide file, printer and other support. Servers do the providing. Redirectors are redirectors, but there are two kinds of server, dedicated and non-dedicated. Dedicated servers just sit there and serve; it's not possible to run any other sort of software on them at the same time. Non-dedicated servers can, in theory, be used as workstations as well. In practice, all the things that users do with impunity to their personal workstations - three-key resets, writing or running dangerous software, changing configurations - can cause serious problems when that machine is also supporting five or six other users. Dedicated servers have another advantage: they can devote everything to the network.

Most network software - such as Novell's NetWare, probably the commonest networking system in use in the UK - allows the choice between dedicated and non-dedicated servers to be made during installation, with the tradeoff between flexibility and performance decided by the installer. The difference can be very significant, since dedicated servers can have their disks formatted in non-standard (for DOS) ways that maximise their performance for networking application, the processor configured for maximum efficiency and MS-DOS completely abandoned in preference for a customised operating system. Indeed, with Novell, you can choose anything from an XT to a 486-based system, or even a VAX as a server. All the compatibility is provided in the network interface.

There are advantages in having non-dedicated, DOS-compatible servers. If everyone has one, then everyone else can share their disk drives, printers or modems at will. Bear in mind, though, that such freedom does bring administrative and security problems; if everyone on your network can use your hard disk, then every program they install can do so too.

Conversely, dedicated servers can provide mainframe levels of security. Every file opened, every message sent, can be logged and the relevant details inspected by the system manager later. More importantly, people can be prevented from altering communal information; you probably don't want everyone being able to change your standard header files, let alone having the privileges to delete the entire system.

Software such as Novell or Nestar's Nex/OS 386 allows very fine control over the access each individual has to disks, directories or individual files. Make sure, however, that your chosen network operating system has the tools to give the system manager easy access to the protection mechanisms - many networks are dangerously insecure merely because nobody on site can understand the incomprehensible, but otherwise entirely adequate, security system.

Management

It's essential to appoint a systems manager for all but the smallest networks, and essential to have them properly trained. A thousand pounds spent on a three-day training course, plus the time lost while that person is out of the office, might seem like a large investment, but compare it to the time lost if five programmers are twiddling their thumbs while the system is down for a day. When computers are networked, the potential for expensive mistakes rises exponentially in respect to the number of machines involved. Wherever you buy your server, make sure you take out a maintenance contract. If the supplier doesn't offer a four-hour callout service, go to someone that does.

Another way of mitigating the dependence a system has on its server, and thus the potential for expensive mishaps, is fault tolerance. Once the province of mainframes, such things as disk mirroring, where a hard disk has each operation duplicated on a separate, independent unit, and power failure management, where loss of mains power leads to a back-up supply kicking in and giving ten minute's grace for a controlled shutdown, are now available on microcomputer networks. Opus Technology provides disk mirroring in its Datasafe PC, and Compaq's latest high-performance servers have similar potential; Netware can also make this available with the right controllers.

Network Versions

Most compilers, assemblers and program maintenance tools don't care whether they're being run from a networked drive by 30 people, or from a local hard disk by one. On a legal point, the licence often cares deeply, and it's worth checking whether you need to buy permission to have your one copy of C installed on the server where ten programmers have access to it. Usually, you do.

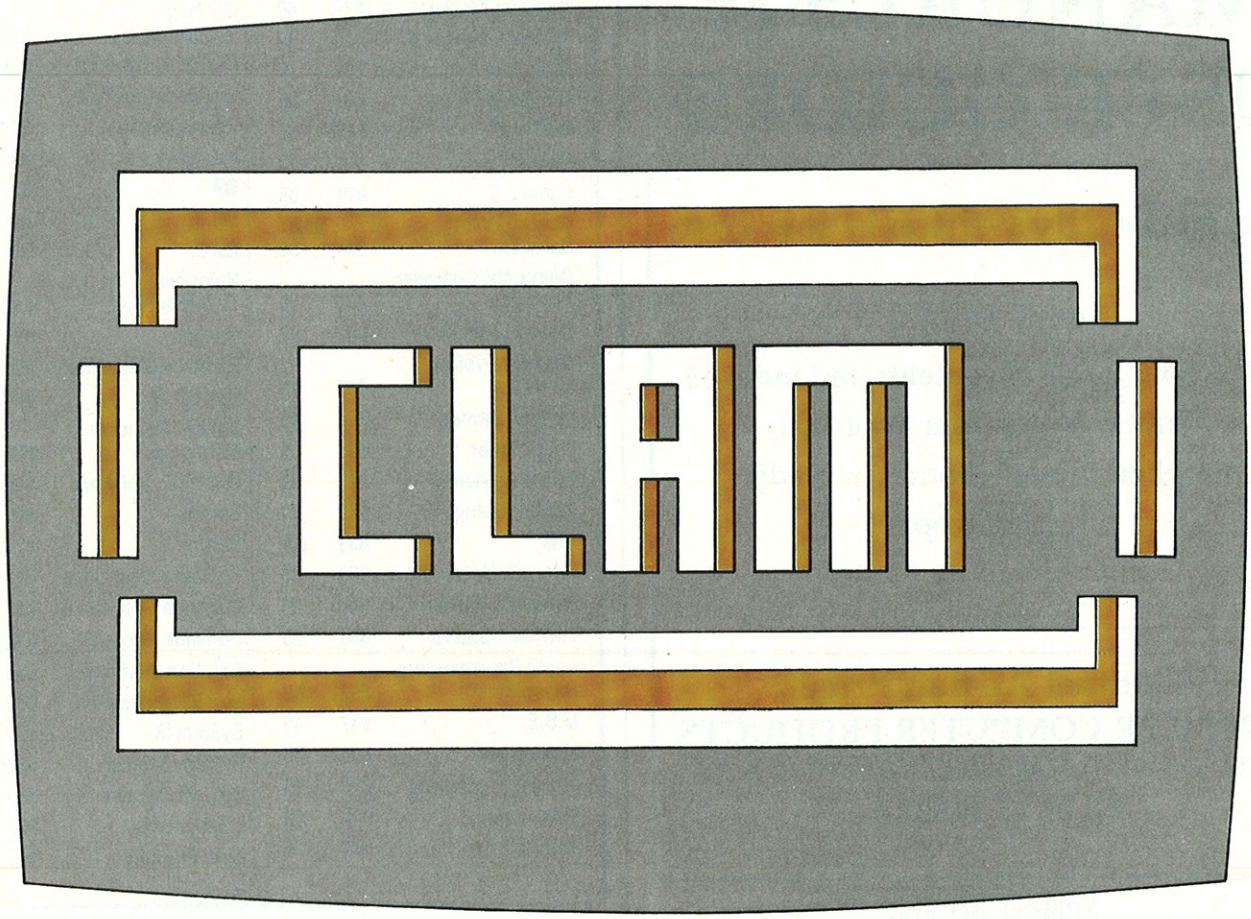
One consideration for developers of networkable software is the interface to which to program. With DOS, if you stick to the published Microsoft network interfacing specifications, you're assured of maximum compatibility but at the expense of performance. OS/2 has networking ingrained at a deeper level, but some understanding of the underlying protocols does make it easier to produce good multi-user software. However, for both these systems, you do have to choose between following the official dictat rigorously and being assured of upwards compatibility, or going for higher performance and chancing a major rewrite when the next version becomes popular.

To some extent, this is changing. Microsoft, IBM and 3Com are now working to the same basic standards, although UNIX internetwork compatibility is still not part of that fold. Novell, by dint of their pre-eminence in the PC networking field, are still able to innovate and stay apart, so it still pays for the network developer to understand and use the system that best meets his needs. .EXE

Rupert Goodwins is with London-based Alfa Systems, software and hardware designers.

Useful Contacts

$25 Network
EQ Consultants
0334 84248
BICC
Suppliers of IsoLan
0442 231000
3Com
3+Open and other products
0628 890670
Compaq
PCs and servers
01 332 3000
IBM
PCs, OS/2 Networking
01 928 1777
Microsoft
OS/2 Lan Manager
0734 391123
Nestar
Makers of Nex/OS
0932 353911
Novell
Netware and Netware/386
0344 860400
Opus
Datasafe PC
0737 765080
Polytron
Version control systems
UK distributor: Software Generation
0462 422525
SPI
The Invisible Network
0734 844081



The definitive data protection software for micros and micro-networks

MAIN FEATURES

- All data held on a hard or floppy disk or a network can be kept secure from unauthorised access.
- Security is by default and is failsafe. Once CLAM has been set up the user does not have to take any positive action to secure data.
- Protected files are kept in a permanently encrypted form on the disk. Even if there is a power or hardware failure the data will never be readable by an unauthorised user.
- A log of all activities can be kept.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 01-948 8255.

£148 + VAT Single User.
£580 + VAT per 10 or part of 10 stations on a network.
Site and Corporate Licences available.

CIRCLE NO. 875

ADVERTISERS INDEX

.EXE APPOINTMENTS

Group Recruitment Sales Manager
Jane Lyons 01-994 6477 x2238

Recruitment Sales Executives
Jeremy Cox 01-994 6477 x2337
Ed Butcher 01-994 6477 x2338

Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH Fax No: 01-994 1533

Rates:

1/4 page	132mm x 88mm	£520
1/2 page	132mm x 179mm	£955
1 page	273mm x 179mm	£1795

Copy Deadline: 18th of the month of publication.

Cancellation: Advertisements may not be cancelled after the copy deadline.

PC Engineer/Technical Support **to £18,000**

Wholly owned subsidiary of worldwide financial information company seeks engineer with knowledge of PC networks, possibly with some programming back-ground. The successful applicant will be ambitious, professional, and will be happy to work as part of a small team on all aspects of system support. Extensive European travel is possible, so a full EEC passport is preferable.

'C' Programmer **to £18,000**

Major communications and telex software manufacturer requires 'C' programmer of at least 1 year's experience to complement existing Research and Development team. The successful applicant will preferably be a graduate, and will definitely be ambitious, conscientious, and a team player. In return the company offers a young, dynamic environment, full product training and excellent prospects for advancement.

Analyst/Programmer **£ negotiable**

RPG II Programmers, Analyst/Programmers with 6 month - 2 years' experience urgently required. We currently have three clients, all committed to AS/400 within 12 months, all willing to retrain to AS/400, and all urgently seeking System 36 skills. Call Terry for immediate interview.

COBOL Programmers- any hardware
to £20,000 + benefits

Major Dealer and Software House requires COBOL programmers for retraining to Ingres. Preferred additional attributes include a good education, UNIX experience, and possibly knowledge of some commercial 4GL. The only essential requirement, however, is a thorough knowledge of COBOL, as full retraining will be provided for the correct applicant.

Programmer 'C'/AIX **to £16,000**

Specialist Software House seeks 'C' programmers with a minimum of 12 months' commercial programming experience under any operating system. Accounts experience will be of particular interest, although again not essential as the company offers full training in addition to generous benefits and structured promotional prospects.

RPG III/COBOL Programmers - AS/400
to £26,000

City based financial information company requires AS/400 Analyst/Programmers, preferably with RPG III on AS/400, although cross training will be provided from system 38. Also of interest is COBOL programming experience on S/3X or AS/400. In all positions a minimum of 18 months solid commercial programming experience is required, possibly within financial environments.

For an immediate interview regarding these and similar current vacancies, call or send/FAX your c.v. to:

Ronicom Recruitment

SEARCH AND SELECTION CONSULTANTS

46-47 Pall Mall, London SW1Y 5JG Tel No: 01-321 0245 Fax No: 01-839 7629

Or if you would prefer to contact us after office hours, call Terry on 02406 5892, or Paul on 01-398 0942

TECHNICAL SOFTWARE

Advanced Graphics

M3 Corridor **to £20K**

Head up displays. Image/Signal Processing and Advanced Graphics Engines. Just a sample of the exciting projects available to you within this expanding graphics consultancy and design house. A rare opportunity to expand your graphics experience within the field of commercial and military visual simulation, image processing and graphic control and display systems. If you have real time or high level languages experience and require technical interest and stimulation in your next career move then this is the ideal opportunity.

Ref: MJ/64

Unlimited Career Progression

Bucks **to £27K**

If you are a creative, ingenious software engineer with 3+ years experience in C, UNIX, VAX/VMS and IBM PC, our client offers you the chance to work on a variety of challenging and stimulating emulation packages. You will be involved from conception through to completion and work on a variety of platforms including state-of-the-art unreleased workstations. The opportunity and package offered by this young ambitious company is second to none. **Ref: MJ/48**

C → C++

Northern Home Counties **to £30K**

Would you like to develop your expertise in Graphics, Database, Data Comms, Object Orientated Design, Expert Systems or Distributed Processing? Would you welcome the opportunity to TRAIN IN C++? If so, and you can offer a good technical degree and a minimum of 1 year's experience in C/UNIX you could be working on a variety of technically challenging projects within a small rapidly growing Systems House. Along with this golden training opportunity my client offers Private Medical Care, Company Pension, Bonus Scheme and Employee Share Options

Ref: MJ/63

Real Time Software Engineers

SW/Home Counties
£18k + Car + Excellent Benefits

We have exciting opportunities within this dynamic Real Time Software/Systems house situated in the SW Home Counties. Market sectors include Communications, Industry and Defence. If you have experience in any of the following:

ADA UNIX SSADM C VMS
YOURDON PASCAL OOD MASCOT

then our client would be very keen to hear form you. Benefits include a car, profit sharing, private health insurance, regular reviews and a team commitment to your personal career development. A first class opportunity for you in the 90's!
Contact: PAUL JONES **Ref: MJ/201**

Secure Computing

Hampshire **to £30K + Car**

Our client, a joint venture company between one of the UK's leading technical consultancies and the UK's foremost communications company are actively seeking experienced software engineers to lead the way in Secure Computing Systems. With good, sound experience in C/UNIX, 68000 ASSEMBLER, 4GL/SQL, FORMAL METHODS or SECURE OPERATING SYSTEMS. You can expect to break new ground in the Planning, Design, Implementation and Evaluation of Total Security Systems for the Government, Industry and Commerce sectors. Technically challenging, stimulating work in an advanced technology environment ensure that all positions from programmers to Consultants enjoy complete career satisfaction in the industry's most exciting growth potential field. A generous salary, relocation, company pension, private medical insurance and BUPA complete the benefits package. **Ref: MJ/65**

Recent Graduates/ MSc Graduates

Various Locations **to £15K**

We have a large number of clients ranging from Major International Defence Contractors through to small Independent Software Houses who have an urgent requirement for technically qualified graduates with software experience. If you can offer C/UNIX, ADA, PASCAL or any Real Time experience we can offer you a number of interesting opportunities.

Ref: MJ/57

For more details on these opportunities call ***Mike Jenkins on 0442 231691*** *office hours or 0582 456417 eves/wkends. Alternatively mail CV to address opposite or fax on 0442 230063*

Executive Recruitment Services

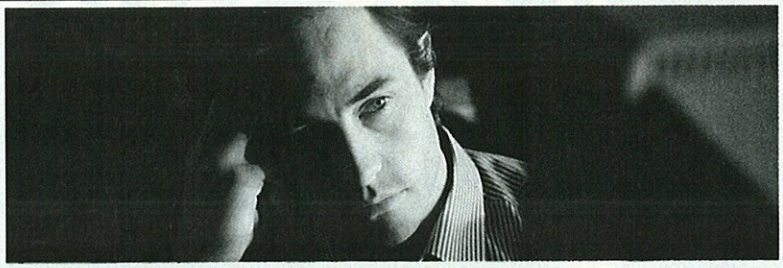
Hempstead House, Selden Hill, Hemel Hempstead, Herts, HP2 4LT

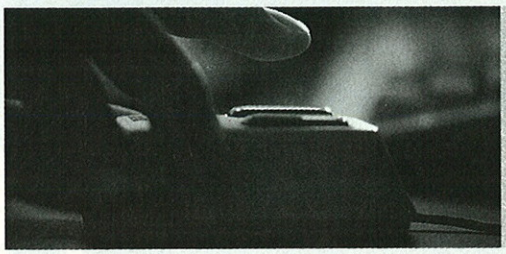
SYSTEM SOFTWARE DEVELOPMENT

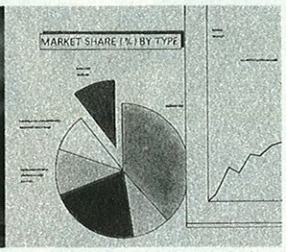
RANK XEROX

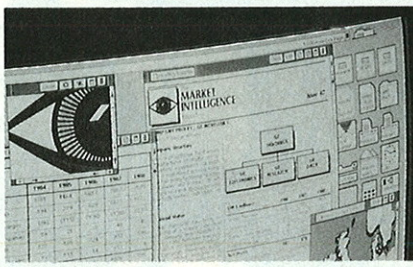
As the System Software Applications Group, developing software products within Rank Xerox multinational office systems business, we are responsible for the design, development and integration of networked document management applications across our own and other manufacturers' systems.

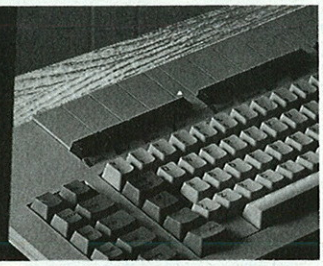
A network of Sun, Macintosh, IBM and Xerox workstations and servers forms the environment for a number of challenging opportunities, as our organisation continues to expand. These include: Networked Desktop/Window/Graphical User Interface applications and tools under Unix and other major operating systems; Document Architectures incorporating text, vector and bit map graphics; Document Interchange; Publishing Systems; Page Description Languages; and Distributed Filing/Printing/Mailing.











Welwyn Garden City

- **Software Engineers**
- **Technical Specialists**

£16K-£33K + car at senior levels

SOFTWARE ENGINEERS and PROJECT LEADERS

You will be of graduate calibre with well developed software engineering skills gained ideally in a Unix and 'C' environment. Experience of MESA, C++ or other object oriented programming languages will be of particular interest.

Whatever your specialisation, Rank Xerox is committed to providing training to ensure that you are equipped to meet our objectives and quality standards.

TECHNICAL SPECIALISTS

The role of our Technical Specialists is to provide in depth consultancy to the members of our development teams. You will enjoy considerable technical responsibility and make major input to our projects as a whole.

You will be of graduate calibre, have sound software engineering skills and at least 2 years experience in your technical specialisation.

TECHNICAL SPECIALIST – UNIX

Your expertise encompasses Unix Kernels and customisation, Unix in a Sun networked environment (TCP/IP, FTP, NFS) and a breadth of understanding of Unix flavours (OSF, UI etc.)

TECHNICAL SPECIALIST – NETWORKS

You will have expert practical knowledge of one or more of the following:

Network Architectures (e.g. OSI, DoD, IBM, XNS, DECNET)

Communication Protocols (e.g. TCP/IP, Datagrams, Appletalk, Ethernet)

Proprietary communication network facilities (e.g. Novell, 3COM).

TECHNICAL SPECIALIST – DOCUMENT PROCESSING

Your background may be in Textual Wordprocessing, WYSIWYG Editors, Graphics toolkits, document creation or printing decomposition.

Your expertise will encompass one or more of: SGML, Document Interchange Formats (e.g. ODIF); PDLs (e.g. Postscript, Interpress); Graphical Contents (e.g. IGES, CGM, RES).

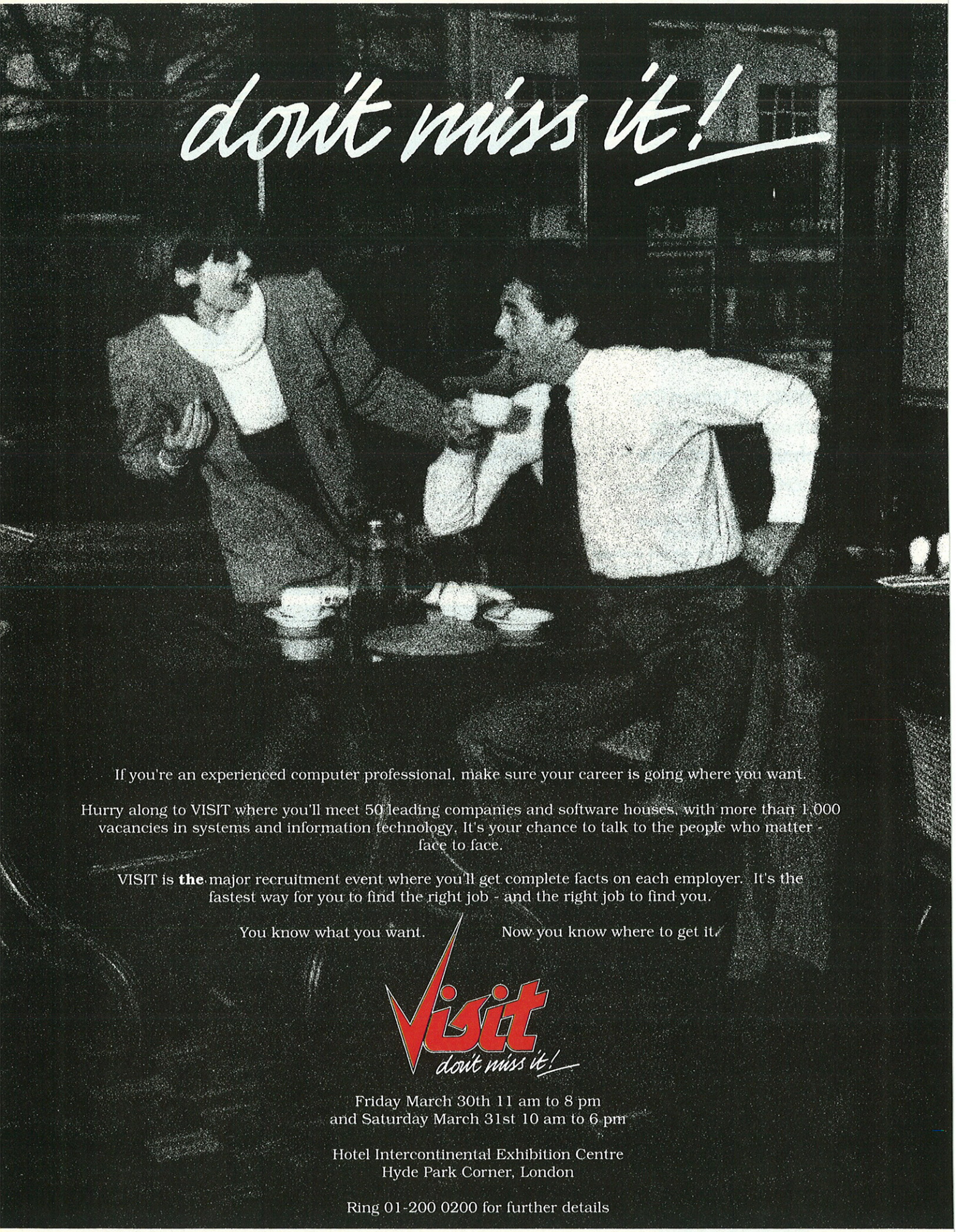
As well as outstanding opportunities and competitive salaries, benefits include BUPA and relocation assistance, where appropriate, to an attractive location.

Rank Xerox is an equal opportunities employer – applications are invited from suitably qualified men and women.

CTA Quest have been retained to provide you with all necessary information about these opportunities so call Julian Goddard or Chris Garrett on 0532 486644 (office hours) or 0423 889742 (evenings/weekends). Alternatively, write to them, enclosing a CV, at the address shown, quoting ref 801/JG.

CTA Quest Recruitment,
Kernel House, Killingbeck Drive,
Leeds LS14 6UF.
Tel: 0532 486644.
Fax: 0532 480105

Initial interviews will be held at CTA Quest premises in Watford, Wokingham and Leeds.



If you're an experienced computer professional, make sure your career is going where you want.

Hurry along to VISIT where you'll meet 50 leading companies and software houses, with more than 1,000 vacancies in systems and information technology. It's your chance to talk to the people who matter - face to face.

VISIT is **the** major recruitment event where you'll get complete facts on each employer. It's the fastest way for you to find the right job - and the right job to find you.

You know what you want. Now you know where to get it.

Visit
don't miss it!

Friday March 30th 11 am to 8 pm
and Saturday March 31st 10 am to 6 pm

Hotel Intercontinental Exhibition Centre
Hyde Park Corner, London

Ring 01-200 0200 for further details

THE AMERICAN WAY

Is a company justified in recouping training costs from staff who leave?

Electronic Data Systems (EDS) is in the news again over its infamous pay-for-your-training policy, having recently won a long-running and bitter legal action it brought against an ex-employee. The company holds the view that if an employee receives training and subsequently leaves the company *a short time afterwards*, then that employee should contribute to the cost of the training. Most good companies send their staff on training courses on a fairly regular basis. Computer training is very expensive, yet most of these firms are prepared to cover the cost in full, regardless of the circumstances. I contacted a number of training companies to find out how prevalent this EDS-type policy is. Although the practice is certainly not widespread, EDS are not alone. What they seem to have in common with other practitioners is that they are a medium/large American company.

Obviously, implementation of such a policy cannot work without knowledge and acceptance by all concerned. Briefly, EDS's own procedure is as follows:

During the interview/selection process a candidate is given ample information about the company, its activities and its policies (including the Systems Engineer Development programme) and what is expected from its staff. No one joins EDS without being made aware of the policy with regard to training on the SED Programme.

The idea is to provide personal support to the individual in order that he or she can become a valuable asset to the company and its clients and, at the same time, develop that person's career. Like most firms, EDS does not like to lose highly-valued personnel, having invested substantial time and money in their development. But unlike most firms, they actually do something about it, thus the pay-back business. The law courts (and several EDS ex-staffers whom I have spoken to) feel that the company is justified. So why don't more firms follow suit?

The only disadvantage I can discern comes through adverse publicity. Reportage is inherently sensational, so that most of what comes out is an image of conflict and repression, the Dickensian employer, dirty laundry in public etc. The best PR cannot compete with a juicy dispute.

The benefits, though, are numerous. First, it helps to eliminate taking on fly-by-night types, those who change jobs every 18 months or so in order to make quantum leaps in salary and promotion; people join the company for the right reasons. Second, what can emerge is a feeling of partnership between employer and employee. 'You are giving us service to enable us to grow our business; we are giving you training to enable you to develop your skills and career.' Very American, but hardly draconian.

One must conclude that either it is felt that the disadvantages outweigh the benefits, or simply that it is not worth all the hassle (this is rather puzzling; it all goes on the balance sheet). Or perhaps it is British mistrust of The American Way. It seems that although many may agree with EDS's stand on this matter, few actually have the mettle to implement such a policy themselves.

Mike Paterson runs UK Appointments, which specialises in finding UK-based work for applicants from abroad. He can be contacted on 081 994 1854

'C' & UNIX SPECIALISTS

Sun Systems Administrator £17–25,000 S. E. London

Our client, a well known computer manufacturer, has a large network of Sun Workstations, primarily used for Electronic Design Automation. This position is within the system support group, providing computer and communication services to their UK and European sites. The ideal candidate should be a self starter with two years Unix/Sun systems administration experience, 'C' programming skills and possibly PC, VAX/VMS or Apollo equipment knowledge. *(Ref PCEX5/1)*

C Analyst/Programmers £15–18,000 London

A leading software house with on-going projects in the UK and Europe has a requirement for high quality C Analyst/Programmers with at least 18 months development experience. A graduate, you will have major project experience, preferably with a real-time based applications bias and the desire to work on both MS DOS and Unix based projects. The London based position will offer the opportunity to work throughout the UK. *(Ref PCEX5/2)*

Unix & C Programmers To £15,000 City

This financial software house is seeking three 'C' and Unix programmers to design and develop applications for a major merchant bank. Preferably a graduate, you will have at least 12 months 'C' under Unix. An excellent remuneration package and career prospects will be offered to the right candidate. *(Ref PCEX5/3)*

Unix & C Development c. £18,000 S. London

Use your 2 years Unix and C experience to develop communications software with this progressive software house. In return for your strong 'C' skills, you'll receive a wide training in products such as Ethernet and X25, and an excellent starting salary. *(Ref PCEX5/4)*

For further details, please contact Suzanne Sadarangani, quoting the appropriate reference number, on 01-734 4010 (office hours) or 01-293 4635 (evenings/weekends). Alternatively, write to her at McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB or fax your C.V. to her on 01-734 1297.

Telephone 01-734 4010

mcgregor boyall

PC Recruitment Specialists

I. T. CAREER APPOINTMENTS

Project Manager
M4/M25 to £27K+car
A strong 'man-manager' with extensive development skills in PC systems, LANS and PC/IBM M/F systems is req. to push on with extensive development of this international datacentre. Ref: EXE 02

PC Programmer
S/Counties to £22K+bens
A solid background in applications development. PC development products and a desire to get involved in analysis are some of the key skills needed by this market leading IT systems house. Ref: EXE 03

UNIX 4GLs
London/S.E. to £23K
Min. 18 mths UNIX with exposure to COBOL, 'C' INFORMIX, ORACLE and CASE tools. S/W House ensures exciting projects. Ref: EXE 07

'C' A/Progs
London to £20K
Min. 2 yrs with 'C' developing POS type systems. Client interaction is a key role in this appointment. Super benefits pkge. Ref: EXE 08

Software Engineer
Kent/Sussex to £19K+
International group requires "client friendly" professionals with 'C' exp in PC/Windowing environment knowledge of real-time multi-tasking systems. Ref: EXE 09

VAX Programmers
S/Counties to £20K
A number of clients have needs for Progs. with 12 mths min. exp. using ORACLE, INGRES, C and 4GLs. Wide spectrum of projects and career opportunities. Ref: EXE 10

For further information about these opportunities, or the many other client needs that we have in London and throughout the Southern Counties telephone Gill McKay – Resources Manager. Alternatively Post or Fax your resume to us.

PROJECT 3
I. T. RECRUITMENT SERVICES LIMITED
Keeley House, 22-30 Keeley Road, Croydon CR0 1TE
Tel: 01 651 3218 (24 hrs) Fax: 01 651 3251

LONDON & SOUTHERN COUNTIES

SOFTWARE DEVELOPERS OXON & BUCKS

PROGRAMMERS - C & ASSEMBLER To £25,000 + Car

This highly successful Software House produces financial dealing room systems using industry standard software including DOS, OS/2 and NET-BIOS. As programmer you will develop leading edge software in a highly integrated operating environment which includes real-time processing, LANs, WANs and communications. Extensive C programming experience is essential. Experience of 286/386 assembly language is desirable.

PC DEVELOPMENT - OS/2 & C £15,000 - £19,000

Superb opportunity to make a significant contribution in a new project group working on the development of a major Retail Distribution system. Responsibilities include systems design, programming and support. Candidates should be experienced C programmers, have exposure to OS/2, PM, and have strong inter-personal skills. Positions also available for candidates with MS-Windows and PASCAL experience.

PC DEVELOPMENT - FOXBASE/dBASE/CLIPPER £15,000 - £22,000

Our client in the Financial Services market continues to make significant investment in the latest PC hardware and software development tools. PC analysts have accountability for the complete development life-cycle including requirements definition, systems design, programming and implementation. Candidates should possess excellent self-presentation and business analysis skills and be fluent in at least one database language.

To discuss your next career move in Oxfordshire and surrounding counties, call David Adcock in strict confidence on 0865-742456 & 7 or send your CV to:

Haybrook Appointments, Suite B, Regal Court, 112 London Road, Headington, Oxford, OX3 9AU

Haybrook APPOINTMENTS

PC Specialists

SOFTWARE ENGINEER TO £20,000 LIVERPOOL
Telecoms Company looking for Graduate Engineers with experience of 'C', COBOL, Assemblers, and some Analogue/Digital design. Excellent benefits package.

SOFTWARE ENGINEER TO £18,000 GUILDFORD
Software house searching for Graduate Engineers with experience of Data capture, Realtime development and Data base design. Varied work within small company.

SOFTWARE ENGINEER TO £26,000 BATH
Superb young software house looking for experienced Graduate engineers with knowledge of 'C', INGRES, PS/2 and SSADM. Excellent benefits package.

SOFTWARE ENGINEER TO £25,000 HANTS
Small company involved in underwater acoustics are looking for a Graduate engineer with experience of High level design, DSP, Networking. Full benefits package.

SOFTWARE ENGINEER £20 PER HOUR HANTS
Large M.O.D Company looking for Engineer with Degree or equivalent in Computer Science 2 years' experience in either, ADA, Coral 66, Pascal RTL2, Fortran, Occam, or either VAX, Intel, Argus, IBM PC, Apollo/Sun, Transputer.

RIGHT ASSOCIATES EMPLOYMENT LTD.
1/6 CITY BUILDINGS
COMMERCIAL ROAD
PORTSMOUTH
HAMPSHIRE, PO1 4BD
TEL: (0705) 293636 FAX: (0705) 756808

PROMOTION WITH UNIX
SUPPORT ANALYSTS

LONDON £20K-£30K + CAR

Young, ambitious Support Specialists with UNIX/XENIX **or** AIX experience sought by **the** leading UK UNIX Consultancy.

Small Co. environment — opportunities in CONSULTANCY, SYSTEM DEVELOPMENT, EDUCATION & TRAINING — technical Variety OS/2, INFORMIX, ALL UNIX VARIANTS, OPEN SYSTEMS . . . and more!

Call Paul Simpson on 01-549 6441.
(6243)

ab executive
HIGH TECHNOLOGY RECRUITMENT
59 Eden Street, Kingston-Upon-Thames, Surrey KT1 1BW. Tel: 01-549 6441 (24-hrs)

FREE VACANCY LIST ALL LOCATIONS

SALARIES £10K – £35K

Simply leave your name and address on our answering machine. Alternatively Post/Fax your C.V.

ABACUS CAREERS LTD

3rd FLOOR, SMITHFIELD HOUSE, DIGBETH, BIRMINGHAM B5 6BS
Telephone: 021-666 7484 Fax: 021-666 7430

software engineering

WEST COUNTRY & SOUTH WALES

Senior/Software Engineers
South Wales
To £17K + Benefits
Rapidly expanding company specialising in voice and data comms require engineers to develop software for ISDN products. Experience of writing and testing C and 68000 under SUN UNIX, or datacomms programming in communication industry would be beneficial.

Software Engineers
South Wales
To £20K + Relocation
Major international technology contractor urgently seeks Real-Time software professionals with experience of PASCAL, CORAL or C under VMS or UNIX.

Software Engineers
South Wales
To £18K + Benefits
Respected telecommunication company seeks engineers with at least 2 years Real-Time, high and low level experience. Preference given to candidates with PABX experience.

Software Engineer
Wilts To £16.5K + Benefits
Leading manufacturer of turnkey computer and communication systems require software professional with 3 years+ VAX programming experience – FORTRAN preferred. DECNET/X25 experience will be beneficial.

Systems Analyst – Team Leader
Wilts To £20K
Excellent opportunity for ambitious engineers with experience of Real-Time FORTRAN, C, PASCAL on VAXes to join major organisation developing Automation and Control systems for the UK and overseas markets.

Application/Test Engineers
Dorset
To £18K + Relocation
Innovative electronics engineers with 12 months experience are required to develop complete Application Test Packages by UK leader.

Software Engineers
Somerset £12-22K
See your career progress with this highly respected organisation who require Real-Time professionals for site work. Experience of PASCAL, CORAL and MASCOT gained in a VAX or embedded microprocessor environment would be ideal.

Software Engineers
South Coast £12-23K
Experience in Real-Time software development using CORAL, PASCAL or C. Suitable candidates will be involved in the adaptive engineering of signalling and switching software products.

Programmer – Senior Analyst/Programmer
Avon To £19K + Benefits
Maths/Physics graduates, at least 12 months FORTRAN under VAX/VMS experience with mathematical modelling techniques required for young software house. Knowledge of professional software development methods & graphics desirable.

Software Engineers
All Areas To £16K
A number of our clients currently require Avionics specialists with PASCAL, CORAL, FORTRAN, C or ADA experience gained in a Real-Time environment, preferably with exposure to VAX or embedded micro-processor hardware.

Programmers to Consultant
Avon £Negotiable
Prestigious software house/consultancy seeks graduates with Real-Time high and/or low level language experience gained in a DEC or IBM environment to provide a professional service to their varied and wide ranging client base.

Analyst/Programmers
Avon To £20K
An expanding systems house currently requires analyst/programmers to work on Real-Time telemetry based systems. Technical requirements include either PASCAL, FORTRAN, C, ADA or CORAL gained within a VMS or UNIX environment.

Software Engineers
Devon To £16K
Two years experience of 'C' essential for this demanding role developing telemetry, high availability and fail-safe systems. Knowledge of VAX and 68000 ideal.

Software Engineers
Glos To £22K
A leading software house require a number of programmers with at least one year's Real-Time experience for challenging roles. Experience of ADA, PASCAL, C/UNIX, ASSEMBLERS or 4GL's are of particular interest, but others will be considered.

Software Engineers
Wilts To £20K
Following their recent expansion, a major engineering company urgently require experience of Real-Time and/or micro-processor systems for challenging design projects.

Programmer/Analysts – Team Leaders
Avon To £20K
Required for software development in rapidly expanding consultancy in VAX/VMS environment. Experience of implementing RDBS for commercial applications using ORACLE or INGRES preferred.

THAMES VALLEY

Software Engineer
Oxon To £18K
Innovative, enthusiastic individual required for a challenging position in R&D. With experience of C, PASCAL and Real-Time, needed to develop systems for clients in Audio/Visual industry.

Software Engineers
Oxon To £19K
2 years+ C, PASCAL on Real-Time Systems required to develop graphics/3D applications in a friendly, creative environment.

Software Engineers
Berks To £23K + Benefits
Leading front-end graphics development, for Software Engineers with experience of C or PASCAL. Knowledge of comms, graphics or interfacing advantageous.

Software Engineer
Wilts To £18K
Systems engineering, graphics environment, 68000, 6809 and C necessary to join a small team in a rapidly expanding young company.

Programmer
Berks To £18K + Benefits
Central systems development, exciting products, a challenging role – 2 years Hi-Level programming on DEC VAX or PDP.

Software Engineers
Berks To £19K
Leading multinational systems house requires solid development experience in any of C, PASCAL or FORTRAN on more than one operating system to develop interconnectivity software.

Programmer
Wilts £14-19K
A Graduate with 2 years+ C and BASIC for development of a leading package for the financial industry. Exposure to graphics an advantage.

Software Engineers
Wilts To £20K + Benefits
Join a young, dynamic systems house, friendly team environment. You will need a good mathematical ability, experience in C and PASCAL under DOS/UNIX, and be looking to expand your analytical and development skills.

Design Engineer
Wilts To £17K
A Graduate with 18 months experience of 'C' under DOS and 8086 require for small company designing management information systems for the transport industry.

Programmer
Wilts To £17K
12 months+ PASCAL on Real-Time systems to develop data correlation systems for major commercial organisations.

Software Engineer
Oxon To £19K
A background in PLC's, C, 68000/Z80 in a Real-Time environment, to join forward thinking systems development centre.

Software Engineer
Berks To £20K
Dynamic company requires ambitious engineers with experience in C, UNIX, ASSEMBLER and DOS for a wide variety of systems design and development applications.

Programmer
Berks £12-19K
Leading Datacomms supplier requires Programmer/Developer with at least 12 months experience of PASCAL and C under UNIX with exposure to X25 or Datacomms. Good career prospects.

Development/Support
Berks £11-18K
A recent graduate, articulate, preferably with experience of financial systems development, to support/develop a leading package for the financial systems market.

Software Engineers – All levels
Berks £12-20K
Software development, communications systems for the public sectors. Experience of FORTRAN, PASCAL or C under UNIX and VMS required, exposure to EXPERT SYSTEMS, WINDOWS or DATACOMMS useful.

Software Engineer
Hants £12-22K
Preferably a graduate, from a Manufacturing or Telecoms background with solid experience of C, PASCAL of ASSEMBLER in a structured environment, to develop Telecoms systems for large, established company.

Software Engineers –
All Areas £12-22K
A large number of our clients require Engineers with experience of Real-Time systems, PASCAL, C, or ASSEMBLER under UNIX or VMS. A wide variety of applications, excellent salaries and career prospects.

Software Engineers
Wilts £12K +
2 years+ experience of ORACLE development to work on financial systems with a rapidly expanding, established systems house.

For further information please contact:

Ian James Sanderson Computer Recruitment
3 Marsh Street Bristol BS1 1RT
Tel (0272) 211741 Fax (0272) 226215

Andy Burgess Sanderson Computer Recruitment
Beaver House, Victoria Road Swindon SN1 3BU
Tel (0793) 644436 Fax (0793) 512217

Sanderson

For vacancies in London and the South East please contact:
David Northmore, Sanderson Computer Recruitment, Swan Centre, Fishers Lane, Chiswick London W4 1RX.
Tel 01 995 8141 Fax 01 995 8143

TECHNICAL MARKETING ENGINEERS

M4 Corridor ... **to 28,000 + car**

Contined outstanding growth in revenue has created these unrivalled opportunities to join the Systems products business unit of this highly successful, multi-billion dollar corporation. They have consistently pushed back the frontiers of innovation through a highly developed entrepreneurial style of operation.

The group that you will be joining, headquartered in the UK, drives and manages the European business through an organisation that is now several hundred people strong in twenty locations.

Key to the success of ongoing design-win programmes, you will provide an authoritative source of expertise in your particular specialisation. Degree qualified in Computer Science or a closely related subject, you will possess a well developed interpersonal style, enabling you to represent the organisation at the highest level within the client base.

Microcomputer Platforms

You will be thoroughly familiar with PC architecture, specifically 386 based hardware, coupled with knowledge of at least one Operating System.

You will provide technical expertise for a range of 386 based Microcomputer Platforms to the OEM client base in Europe. This role has a strong commercial flavour and provides the opportunity to get involved with a real architectural influencer!

Unix Workstations

You will be something of a "guru" in the contemporary UNIX environment, with experience in X-Windows or Graphics user interfaces. In addition you will be familiar with traditional hardware solutions in the workstation market.

Your responsibility will be to provide technical focus and direction in the UNIX and Workstation product lines. Some awareness of workstation networking in this role would be an advantage.

TSI Group
QUALITY IN SERVICE

If you are motivated at the prospect of joining and representing a world leader and providing an unrivalled level of technical expertise, then please call **Sybil Temudo** or **Grant Revan** today quoting Ref SDNJ488 on **(0344) 860919**

or write to them at: **TSI Group, The Columbia Centre, Market Street, Bracknell, Berkshire, RG12 1PA,** or fax details on **(0344) 860581**

Information Technology Recruitment Consultants

"C" and MS-WINDOWS PROGRAMMERS
up to £21K + profit share bonus

SUPERB PC SOFTWARE DEVELOPMENT OPPORTUNITIES

As one of the UK's most successful IT consultancies, we are committed to attaining the highest standards of development – both in terms of software and the skills of our staff.

This philosophy has enabled us to build a team of outstanding software development professionals, who combine creative and keen minds with an enthusiasm to learn from colleagues. If you have these attributes you will benefit from exposure to the very latest tools and techniques as well as from more formal training.

We now have opportunities for Senior Programmers with at least two years' experience of "C" – 12 months of which must include development in MS-Windows. While we are happy to train you, knowledge of SQL, CASE, OS/2, Methods, Networks and/or Database Servers would be a positive advantage. We particularly seek a high level of technical ability and the desire to join teams facing some of the most professionally challenging development projects.

Immediate rewards include a salary of up to £21,000 plus a profit share bonus and other benefits including private health care and relocation assistance where appropriate. This is an outstanding opportunity to join the UK's leading developer of CASE products providing superb career development opportunities which will include either software engineering specialisation, team leadership or indeed, a mixture of the two.

For further details, please contact David Drysdale on 01-636 4213 (24 hours). Alternatively, write with your career and salary details to him at Learmonth & Burchett Management Systems Plc, Evelyn House, 62 Oxford Street, London W1N 9LF.

LBMS

STOB - Is it not yet 5 o'clock?

Ms Stob is having difficulty concentrating on her work.

Oh it's you. It's all right, you can come in. I'm just polishing off the selection-of-the-printer-driver module. Look: you enter here with a selector value, index into an array of pointers, call the result, loop round until you get a legal value, adjust the linked list, back out in reverse order and return the file handle to caller. The design is stunningly elegant, although I say so myself; it's user-friendly, it's versatile, it's as cohesive as smoke, it's altogether Verity. The one, tiny problem is: I can't see my way to coding the sod.

Normally, you understand, we would suss this beauty in less time than it takes OS/2 to repaint the small piece of blank screen that was underneath a window that you have just re-sized - ie about one subjective hour. But today I just don't seem able to get down to it. Look at this. I've got to declare an array of pointers to functions returning a pointer to arrays of functions, and I can't even think of a sensible identifier name, let alone work out the syntax. Let's just call it `PooDrops` to be going on with. Hold on, it's more difficult than that, it's got an argument as well, I've got it written down, oh yes, its argument is a pointer to a constant of type `Sheep`.

God, I'm bored. I'm so bored, it feels like Sunday afternoon. Another cup of coffee is in order. Perhaps the injection of a little excitement, in the form of a brief trip to the kitchen, with the promise of a brief trip to the loo 20 minutes later, would put the zest back into my programming style. Or perhaps not. I'm already rigid with caffeine; if I have any more coffee, I'll metamorphose into a coff.

Leaving aside the question of `(*PooDrops[i])(pBaaBaa)`, how about sorting this loop. I want it to run for `(int)BigSpotOnStevesNose` iterations, starting at -17, unless the operator ESCapes or F1s for help. I've got this brilliant method of avoiding fence post looping errors, by counting out these matches on the desk. Loop from -17 to 24. Here we go: -17, -16, -15, -13, sorry, *-14*, -13 again... Hold on, am I supposed to be counting matches, or gaps between matches?

It's only 2.41pm. You'd think that it would be better after lunch, but it isn't. You'd think that after all the effort of struggling up to '12', the little hand of the Swatch would just coast down to '6', no sweat. Oh look, outside in the street. Here comes a man, eight months pregnant and the suspected father must be Bass Charrington, with a pneumatic drill. His squire following with the inevitable compressor at five paces. Doubtless they come to add another Tarmac varicose vein to the impressive network that spans our road. This is good news. At least I shall have a legit excuse for being unable to do any work, not to mention something to watch until home time.

Ah ha! I remember now. It's not a pointer to type `Sheep`, that's just the name of Susie's new boyfriend: Roger Sheep. I wrote it down, to prove to myself that I hadn't made it up. I knew I wouldn't define a type called `Sheep`. Here we are. The argument is really a pointer to type `Dandruff`. So I've got to call `(*PooDrops[i])(pFlakeyBit)` with -17, -16, -14...

.EXE

CANDIDATES CAN'T YOU 'C' THAT WHAT U-NIX IS ALR

MANAGEMENT CONSULTANTS
(Basics £35k-£75K + Exc Bens)

Our clients, one of Europe's largest Software houses/Service bureau and consultancies, seek several Management Consultants. Essential requirements all candidates:
Previous/current Consultancy exp with a consultancy.

- Current salary package exceeds £35K.
- iT B'ground (not necessarily including Prog'g).
- Must have sales exp. (pre/post sales) Must want to undertake an active sales role with the company.
- Must want to be a Manager, though accepting a Consultant role initially and have to earn the promotion.
- Must view the company as a positive career move.
- Must have specific Business Market expertise either:
  (1) Business-Banking, Insurance, Public Sector.
  (2) Application-Payroll, Accounting, Investment Fund Management.

In conclusion, Candidates will be given a full company profile on Interview.

iT CONSULTANTS
(30K-£45K + Exc Bens)

Dynamic Consultancy seek iT Consultants (2-4) with preferred Consultancy exp, previous/current sales exp, 5 years' exp. Areas of expertise req'd are:- Data Mng't with knowledge of RDBM's including DB2, ORACLE, INGRES or Comms Data and Voice, will accept Data only, but not Voice.

BUSINESS CONSULTANTS
(SALARY TO £40K + Exc Bens)

Our client, a Major Independent Software House, seek 6-14 Business Consultants, based City. Must have DP B'ground and still be able to manage projects. Business Analysts/Analysts with exp of Non-Retail Banking (exclud'g Dealing Room Front Office Appli), B.Societies, Insurance, Central or L/Govt (esp Housing Bens), Public Utilities (Good Sales Contacts), Transport, Property Management, GIS (Graphics Info Sys-Digital Mapping), knowledge of Info Plann'g, Data Conversion, Digital Mapping Techniques. In conclusion, Candidates must have excellent interpersonal skills.

GROUP LEADER - UNIX
(C£28k + BENS + CAR)

Our clients, a major Computer Manu'f, seek a Unix Group Leader to manage Group coupled with Tech Leadership. Must be able to meet Benchmarks on time and within budget. Liaison with overseas parent plus responsible for Recruitment and resourcing. Min 5yrs in depth Unix exp with Mng't B'ground. Network skills plus Multi User environment exp desirable. 'C' - TURBO 'C' or RDBMS helpful. Any version Unix will be considered.

In conclusion we have many current DP Vacancies covering Designers, Progs, AP's, Proj Mng's, Bus Analysts, DEC RDB/CDD + Admin, ICL Data Dict'y Sys Admin, Sys Mngr Dec, NetworkSupport Manager, Snr Progs, 'C' Progs/Analysts. Salaries from £12.5K-£38K + Bens. B'grounds interested in IBM M'Frames, IBM AS400, ICL, UNISYS, WANG PACE, ORACLE, HP, 'C', BULL DPS 7/8. Min 1 and a half year's exp.

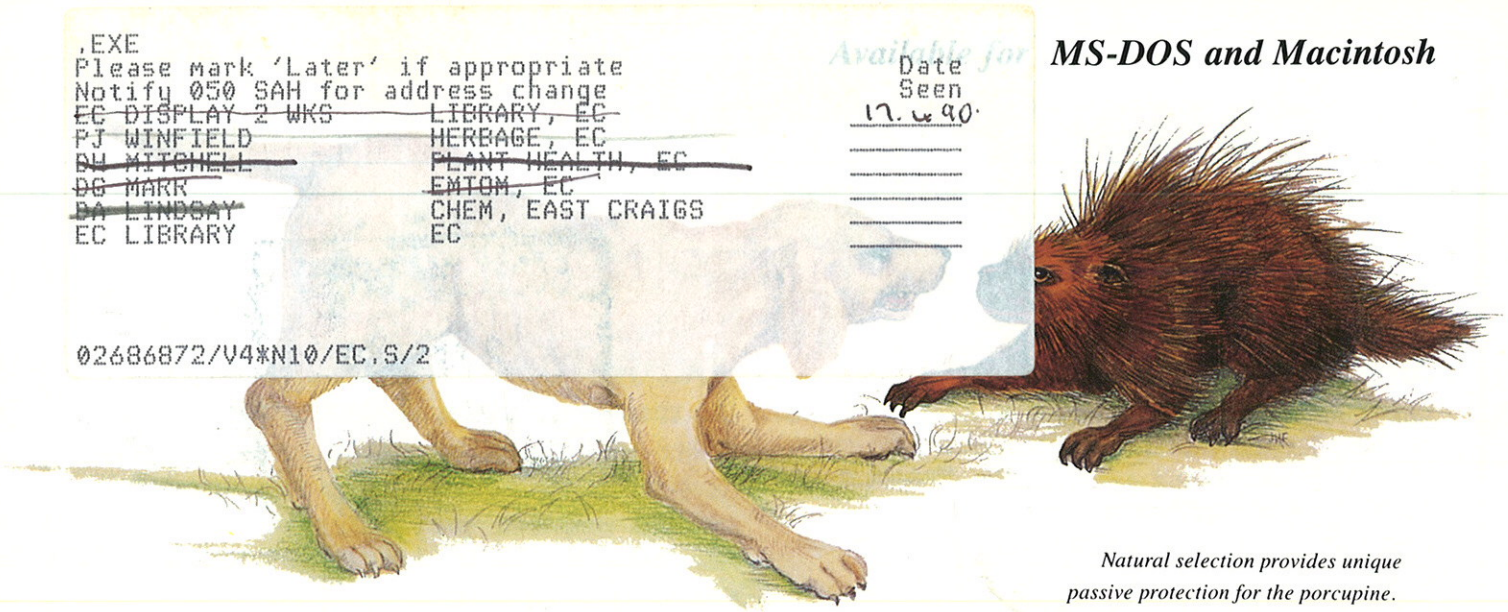
The Consultancy that puts you and your career first - Why compromise?

ALR

ADVANCED LOGIC RECRUITMENT LTD.
Management & Recruitment Consultants

For further information about these openings, please contact us on 01-388 2339/2338 or 01-388 4799 up to 7.30p.m. weekdays or send your cv to us in confidence at the following address:

Euston House, 81-103 Euston Street, London NW1 2ET.



MS-DOS and Macintosh

Natural selection provides unique passive protection for the porcupine.

The Activator - Natural Selection For Software Protection

Inventor and entrepreneur Dick Erett explains how "The Activator" provides sane protection for your intellectual property.

"In any industry, just as in nature, the process of natural selection raises one solution above another. Natural selection is the most elegant of engineers.

In the area of software protection The Block has been selected by the marketplace as the solution that works. Over 500,000 packages are protected by our device.

For the past 4 years our philosophy has been; *'You have the right and obligation to protect your intellectual property.'*

A New Ethic For Software Protection

In allowing end-users unlimited copies of a software package and uninhibited hard disk and LAN operation, The Block has created a new ethic for software protection.

By removing protection from the magnetic media we remove the constraints that have plagued legitimate users.

They simply attach our key to the parallel port and forget it. It is totally transparent, but the software will not run without it.

A New Technology For Software Protection

Our newest model, The Activator, builds on our current patented design, and establishes an unprecedented class of software protection.

We have migrated and enhanced the circuitry of The Block to an ASIC (Application-Specific Integrated Circuit) imbedded in The Activator.

This greatly improves speed and performance, while reducing overall size. Data protection can also be provided.

Programmable Option

The Activator allows the software **developer** the option to program serial numbers, versions, or other pertinent data known only to the developer, into the circuit, and access it from the program.

Once you program your part of the chip, even we have no way to access your information.

The ASIC makes emulation of the device virtually impossible. It also presents an astronomical number of access combinations.

Full 100% Disclosure

Since The Activator is protected by our patent we fully disclose how it works. Once you understand it, endless methods of protection become evident.

Just as no two snowflakes are the same, no two implementations of The Activator are identical. And like the snowflake the simplicity of The Activator is its greatest beauty.

We never cramp your programming style or ingenuity. Make it as simple or complicated as you desire.

Let us help safeguard what's rightfully yours. Please call today for additional information or a demo unit. *It's only natural to protect your software.*"

"On-the-fly" programmable memory option now available for OS/2, MS-DOS and Macintosh

Software Security UK LTD
21A The Precinct, High Street, Egham
Surrey TW20 9PN
TEL: (0784) 430060
FAX: (0784) 430050

Unlimited Copies • Programmable • No Batteries • Small Size • Fast • Patented • Data Protection

CIRCLE NO. 878